

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«На правах рукопису»
УДК 004.422.8::001.18

«До захисту допущено»
Науковий керівник кафедри
_____ І.А. Дичка
«__» _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 Інженерія програмного забезпечення

**на тему: «Метод і програмний засіб прогнозування курсу
криптовалюти»**

Виконав:

студент II курсу, групи КП-71мн
Яхін Сергій Леонідович _____

Керівник:

Доцент кафедри ПЗКС, к.т.н., доцент
Цуркан Василь Васильович _____

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н.,
Онай Микола Володимирович _____

Рецензент:

Доцент кафедри АУТС, к.т.н.,
Дорогий Ярослав Юрійович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою

Спеціальність – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ І.А. Дичка

«___» _____ 2017 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Яхіну Сергію Леонідовичу

1. Тема дисертації «Метод і програмний засіб прогнозування курсу криптовалюти», науковий керівник дисертації Цуркан Василь Васильович, к.т.н., доцент, затверджені наказом по університету від «8» квітня 2019 р. №1075-С
2. Термін подання студентом дисертації «17» травня 2019 р.
3. Об'єкт дослідження: процес прогнозування курсу криптовалюти.
4. Предмет дослідження: метод і програмний засіб прогнозування курсу криптовалюти.
5. Перелік завдань, які потрібно розробити:
 - провести аналіз існуючих методів та практик прогнозування курсу криптовалюти;
 - розробити метод оптимізуючий процес прогнозування курсу криптовалюти;
 - розробити концептуальну модель програмного засобу прогнозування курсу криптовалюти;
 - розробити об'єктно-орієнтовану модель програмного засобу прогнозування курсу криптовалюти;
 - обрати форму репрезентації програмного засобу прогнозування курсу криптовалюти;
 - обрати стек технологій для реалізації програмного засобу прогнозування курсу криптовалюти;
 - розробити програмний засіб прогнозування курсу криптовалюти;
 - дослідити реалізований програмний засіб прогнозування курсу криптовалюти на ефективність.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу:
 - структурна модель методу в нотації IDEF0;
 - декомпозиція структурної моделі методу в нотації IDEF0;
 - декомпозиція структурної моделі методу в нотації IDEF3;

- декомпозиція структурної моделі методу в нотації DFD;
- структурна схема варіантів використання програмного засобу;
- діаграма класів програмного засобу;

7. Орієнтовний перелік публікацій:

- Тези доповіді «Аналіз програмних засобів прогнозування курсу криптовалют»;
- Стаття «Концептуальна модель програмного засобу прогнозування курсу криптовалют»;
- Тези доповіді «Об'єктно-орієнтована модель програмного засобу прогнозування курсу криптовалюти»;
- Тези доповіді «Програмний засіб прогнозування курсу криптовалют».

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., доцент кафедри ПЗКС		

9. Дата видачі завдання «11» жовтня 2017 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Грунтовне ознайомлення з предметною галуззю	15.01.2018	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	14.03.2018	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	13.05.2018	
4.	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення; підготовка матеріалів доповіді на конференції ПМК-2018-2	17.10.2018	
5.	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження	15.12.2018	
6.	Проведення наукового дослідження; робота над третім розділом магістерської дисертації	21.02.2019	
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка ілюстративного матеріалу	25.04.2019	
8.	Оформлення текстової і графічної частини магістерської дисертації	13.05.2019	

Студент

С.Л. Яхін

Науковий керівник дисертації

В.В. Цуркан

РЕФЕРАТ

Актуальність теми. Прогнози стали частиною сучасного світу. Економіка складається в тому числі з очікувань і прогнозів: політичної обстановки, появи та популярності технологій, врожаю на аграрному секторі, погоди, поточної ситуації з видобутком корисних копалин для різноманітних галузей, та прогнозів вартості різноманітних активів та валют. Щоб ефективно працювати на ринку необхідно мати релеванту інформацію, своєчасні прогнози для передбачення майбутнього стану речей, котирування того, чи іншого товару.

І обираючи найвигідніший товар для прогнозування особливим чином виділяються криптовалюти. На сьогоднішній день їх існує понад 2500, сім із них мають капіталізацію в мільярди доларів, сотні інших – в сотнях мільйонів доларів.

Користувацькі впровадження різних криптовалют злетіли: мільярди в ринковій капіталізації і мільйони гаманців, за оцінками, були «активними» в 2016 році. А через їх волатильність – на криптовалютних біржах прибутковість від угоди може досягати тисячі відсотків.

Отже інноваційність, великі перспективи та зацікавленість світової спільноти робить криптовалюти цінним активом, який варто прогнозувати.

Об’єкт дослідження: процес прогнозування курсу криптовалюти.

Предмет дослідження: метод і програмний засіб прогнозування курсу криптовалюти.

Мета роботи: створити метод прогнозу курсу криптовалюти, реалізувати його у вигляді програмного засобу та дослідити ефективність.

Метод дослідження: розгляд і аналіз методів прогнозування курсу валют як Марковські випадкові процеси та ланцюги Маркова, метод експоненціального згладжування з поєднанням результатів за допомогою багатошарового перцептрона Румельхарта.

Наукова новизна полягає в наступному: існуючі методи технічного аналізу (ТА) мають обмежені результати точності прогнозу, та можуть бути ефективно застосовані лише експертами у поєднанні з методами фундаментального аналізу. Даний метод має на меті заміну фінального підсумку прогнозу людиною на прогноз від штучної нейронної мережі (ШНМ) з архітектурою багат шарового перцептрона Румельхарта, яка діятиме на основі даних наданих методами ТА.

Практична цінність отриманих в роботі результатів полягає в тому, що запропонований метод прогнозування курсу криптовалют на основі поєднання результатів методів ТА у єдиний прогноз дозволив зменшити ризики та полегшити недосвідченим користувачам процес торгів на криптовалютній біржі, завдяки чому збільшився їх дохід.

Здійснено програмну реалізацію запропонованого методу, що може бути використана для прогнозування курсу криптовалюти.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018-2 (м. Київ), модель описана в статті «Концептуальна модель програмного засобу прогнозування курсу криптовалюти» у виданні *Моделювання та інформаційні технології*: збірник наукових праць у 2019 році.

Опис програмного засобу прогнозування курсу криптовалюти було продемонстровано на XXXVII науково-технічній конференції молодих вчених та спеціалістів інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України: (м. Київ, 15 травня 2019 р.).

Об'єктно-орієнтована модель програмного засобу прогнозування курсу криптовалюти були описані в тезах та будуть представлені на конференції «Актуальні питання застосування спеціальних інформаційно-телекомунікаційних систем»: (м. Київ, 22 – 23 травня 2019 р.).

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів, висновків та додатків.

У вступі надано загальну характеристику роботи, виконано оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень.

У першому розділі розглянуто існуючі методи для вирішення задач прогнозування курсу валют та криптовалют, які сьогодні використовують трейдери та спеціалізовані програми для аналізу ринку; проведено їх детальний аналіз щодо зручності, часозатрат, типу даних та ефективності прогнозу.

У другому розділі запропонований оптимізований метод прогнозування і розглянуто структурну модель програмного засобу прогнозування криптовалюти на основі запропонованого методу у нотації IDEF0, та її декомпозиціях у нотаціях IDEF0, IDEF3 і DFD.

У третьому розділі запропоновано об'єктно-орієнтовану модель програмного засобу прогнозування курсу криптовалюти з описом діаграми варіантів використання, логічної та фізичної моделі.

У четвертому розділі проаналізовано вимоги до існуючих засобів розробки програмного забезпечення для реалізації методу. Обрано та обґрунтовано вибір стеку технологій, та описано як за його допомогою було створено програмний засіб прогнозування криптовалют. Отриманий засіб було протестовано на працездатність та ефективність.

У висновках проаналізовано отримані результати роботи.

У додатках наведено фрагменти програмної реалізації запропонованого методу та копії графічних матеріалів.

Робота виконана на 70 аркушах, містить 2 додатки та посилання на список використаних літературних джерел з 24 найменувань. У роботі наведено 19 рисунків та 12 таблиць.

Ключові слова: Штучна нейронна мережа, багатошаровий перцептрон, прогнозування курсу, точність прогнозу.

ABSTRACT

Actuality of theme. Forecasts have become part of the modern world. The economy also includes expectations and forecasts: the political situation, the emergence and popularity of technologies, the harvest in the agrarian sector, the weather, the current situation with the extraction of minerals for various industries, and forecasts of the value of various assets and currencies. In order to work effectively in the market, it is necessary to have reliable information, timely forecasts for predicting the future state of things, quotes for a particular product.

And choosing the most advantageous goods for forecasting in a special way stand out cryptocurrency. To date, there are over 2500 of them, with seven of them capitalizing on billions of dollars, hundreds of others – in hundreds of millions of dollars.

Customized implementations of various cryptocurrencies skyrocketed: billions in market capitalization and millions of purses were estimated to be "active" in 2016. And because of their volatility – on cryptocurrency markets profitability from the transaction may reach thousands of percent.

Consequently, the innovation, the great prospects and interest of the world community makes the cry a cheap asset, which is worth to predict.

Object of research: the process of forecasting the course of cryptocurrency.

Subject of research: method and software for forecasting the course of cryptocurrency.

The purpose of the work: to create a method for forecasting the course of cryptocurrency, to implement it as a software tool and to investigate the effectiveness.

Research method: review and analysis of methods for forecasting the exchange rate as the Markov random processes and Markov chains, the method

of exponential smoothing with the combination of results with the help of a multi-layer perceptron of Rumelhart.

The scientific novelty is as follows: existing technical analysis techniques (TA) have limited results of prediction accuracy, and can only be effectively applied by experts in combination with fundamental analysis methods. This method is intended to replace the final result of a human prognosis with an artificial neural network (ANN) prediction with the architecture of the multi-layer perceptron of Rumelhart, which will operate on the basis of data provided by TA methods.

The practical value of the results obtained in the work is that the proposed method of forecasting the exchange rate cryptocurrency on the basis of a combination of results TA methods in a single forecast allowed to reduce the risks and facilitate the inexperienced users bidding process on the cryptocurrency market, which increased their income.

Approbation. The main provisions and results of the work were presented and discussed at the scientific conference of masters and postgraduates "Applied Mathematics and Computer", PMK-2018-2 (Kyiv), the model is described in the article "Conceptual model of the software for forecasting the cryptocurrency course" in the edition Simulation and informational technology: a collection of scientific works in 2019.

A description of the software for forecasting the course of cryptocurrency studies was demonstrated at the XXXVII scientific and technical conference of young scientists and specialists of the Institute of Modeling Problems in Energy. G. E. Puhov NAS of Ukraine: (Kyiv, May 15, 2019).

Object-oriented model of the software for forecasting the course of cryptocurrency were described in theses and will be presented at the conference "Topical issues of application of special information and telecommunication systems": (Kyiv, May 22-23, 2019).

Structure and scope of work. The master's thesis consists of an introduction, four chapters, conclusions and appendices.

The introduction provides a general description of the work, an assessment of the current state of the problem is performed, the relevance of the research direction is substantiated.

In the first section examines the existing methods for solving the prediction of the currency and cryptocurrency currently used by traders and specialized programs for market analysis; Their detailed analysis on convenience, time expenditures, type of data and forecast efficiency was conducted.

In the second section an optimized prediction method is proposed and the structural model of the crypto-prediction software tool based on the proposed method in the IDEF0 notation and its decompositions in the notations IDEF0, IDEF3 and DFD are considered.

In the third section an object-oriented model of the software for predicting the course of cryptocurrency with the description of the diagram of the use variants, the logical and physical model is proposed.

The fourth section analyzes the requirements for existing software development tools for implementing the method. The choice of technology stack has been chosen and grounded, and it is described how it created a software tool for predicting cryptocurrency. This tool was tested for performance and efficiency.

The annexes show fragments of the software implementation of the proposed method and copies of graphic materials.

The work is done on 70 sheets, contains 2 attachments and a link to the list of used literary sources of 24 titles. The paper presents 19 drawings and 12 tables.

Key words: artificial neural network, multilayer perceptron, forecasting of the course, accuracy of the forecast.

ЗМІСТ

ТЕРМІНИ	4
ВСТУП	8
1. АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ	10
1.1. Прогнозування курсу криптовалют	10
1.2. Методи прогнозування курсу криптовалют	11
1.3. Програмні засоби прогнозування курсу криптовалют	18
1.4. Висновки.....	25
2. МЕТОД ТА СТРУКТУРНА МОДЕЛЬ ПРОГРАМНОГО ЗАСОБУ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ.....	26
2.1. Постановка задачі	26
2.2. Новий метод прогнозування курсу криптовалют	26
2.3. Структурна модель програмного засобу	29
2.4. Декомпозиція концептуальної моделі програмного засобу в нотації IDEF0	31
2.5. Декомпозиція концептуальної моделі програмного засобу в нотації IDEF3	33
2.6. Декомпозиція концептуальної моделі програмного засобу в нотації DFD	34
2.7. Висновки.....	36
3. ОБ'ЄКТНО-ОРІЄНТОВАНА МОДЕЛЬ ПРОГРАМНОГО ЗАСОБУ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ.....	38
3.1. Постановка вимог	38
3.2. Діаграма варіантів використання.....	42
3.3. Логічна модель.....	46
3.4. Фізична модель	51
3.5. Висновки.....	55
4. ПРОГРАМНИЙ ЗАСІБ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ	57

4.1. Аналіз вимог до інструментів для реалізації програмного засобу	57
4.2. Обрання платформи та стеку технологій для реалізації програмного засобу	60
4.3. Реалізація програмного засобу	62
4.4. Тестування розробленого програмного засобу	64
4.5. Висновки	67
ВИСНОВКИ	68
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	69

ТЕРМІНИ

- Криптовалюта (КВ, від англ. *Cryptocurrency*) – вид цифрової валюти, емісія та облік якої засновані на асиметричному шифруванні і застосуванні різних криптографічних методів захисту, таких як Proof-of-work та/або Proof-of-stake. Функціонування системи відбувається децентралізовано в розподіленій комп'ютерній мережі.
- Блокчейн, тобто ланцюжок блоків транзакцій (БЧ, англ. *Blockchain, Block chain*, від *block* – блок, *chain* – ланцюг) – розподілена база даних, що зберігає впорядкований ланцюжок записів (так званих блоків), що постійно довшає. Дані захищено від підробки та спотворення. Кожен блок містить часову позначку, геш попереднього блока та дані транзакцій, подані як геш-дерево. Таку розподілену базу даних закладено в основу криптовалют, де слугує бухгалтерською книгою для всіх операцій.
- Майнінг (від англ. *mining*) – діяльність з підтримки розподіленої платформи і створення нових блоків з можливістю отримати винагороду в формі емітованої валюти і комісійних зборів у різних криптовалютах. Вироблені обчислення потрібні для забезпечення захисту від повторного використання одних і тих же одиниць валюти, а зв'язок майнінгу з емісією стимулює людей витрачати свої обчислювальні потужності і підтримувати роботу мереж.
- Біткоїн (англ. *Bitcoin, BTC*) – електронна валюта, концепт якої був озвучений 2008 року Сатоши Накамото, і представлений ним 2009 року, базується на самоопублікованому документі Сатосі Накамото. Bitcoin не має централізованого управління та емітентів. Транзакції із цифровим підписом між двома вузлами передаються до всіх вузлів мережі peer-to-peer, а самі дані про переміщення коштів зберігаються у розподіленій базі даних. Для запобігання можливості втрати чужих грошей або використання своїх коштів двічі використовуються криптографічні методи.

- Криптовалютна біржа (КБ) – майданчик для торгівлі та обміну одних цифрових грошей на інші або ж на різні світові валюти (USD, EUR, RUR, CNY). Крім майнінгу, який є початковим способом створення криптовалюти, біржі є способом їх отримання. Крім цього тільки за допомогою бірж можна обміняти їх на реальні гроші.
- Курс криптовалюти – вираження ціни однієї валютної одиниці в одиницях іншої. Залежить від популярності криптовалюти, реклами і новин. Водночас від поширення криптовалюти і великих гравців, які регулярно практикують памп на біржах.
- Прогнозування курсу криптовалюти – процес визначення майбутнього курсу криптовалюти на основі аналізу її минулого і сучасного.
- Криптотрейдинг (КТ) – один з основних способів заробітку на криптовалютах. Торгівля токенами криптовалюти несуттєво відрізняється від трейдингу цінними металами, валютами та акціями. Основне завдання залишається колишнім: купити якомога дешевше, продати дорожче і отримати прибуток.

Основною відмінністю трейдингу криптовалютою від фондового ринку є висока волатильність курсу. Багато трейдерів мріють про прибутковість в тисячі відсотків від угоди, але на фондовому ринку такої угоди доводиться чекати роками, викуповуючи акції маловідомих компаній в очікуванні зростання їх капіталізації. У той же час швидкі угоди з високою прибутковістю абсолютно не рідкість для криптотрейдингу.

- Часовий ряд (ЧР, англ. *time series*) – це ряд точок даних, проіндексованих (або перелічених, або відкладених на графіку) в хронологічному порядку. Найчастіше часовий ряд є послідовністю, взятою на рівновіддалених точках в часі, які йдуть одна за одною. Таким чином, він є послідовністю даних дискретного часу. Прикладами часових рядів є висоти океанських припливів, кількості сонячних плям, та щоденне середньозважене значення індексу ПФТС на момент закриття торгів.

- Регресійний аналіз (РА) – розділ математичної статистики, присвячений методам аналізу залежності однієї величини від іншої. На відміну від кореляційного аналізу не з'ясовує чи істотний зв'язок, а займається пошуком моделі цього зв'язку, вираженої у функції регресії.
- Метод зворотного поширення помилки – метод навчання багатошарового перцептрону. Це ітеративний градієнтний алгоритм, який використовується з метою мінімізації помилки роботи багатошарового перцептрону та отримання бажаного виходу. Основна ідея цього методу полягає в поширенні сигналів помилки від виходів мережі до її входів, в напрямку, зворотному прямому поширенню сигналів у звичайному режимі роботи. Барц і Охонін запропонували відразу загальний метод («принцип подвійності»), який можна застосувати до ширшого класу систем, включаючи системи з запізненням, розподілені системи.
- Штучні нейронні мережі (ШНМ, англ. *artificial neural networks*, ANN), або конективістські системи (англ. *connectionist systems*) – це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин. Такі системи навчаються задач (поступально покращують свою продуктивність на них), розглядаючи приклади, загалом без спеціального програмування під задачу.

ШНМ ґрунтується на сукупності з'єднаних вузлів, що називають штучними нейронами (аналогічно до біологічних нейронів у головному мозку тварин). Кожне з'єднання (аналогічне синапсові) між штучними нейронами може передавати сигнал від одного до іншого. Штучний нейрон, що отримує сигнал, може обробляти його, й потім сигналізувати штучним нейронам, приєднаним до нього.

В можливості ШНМ в тому числі входить наближення функцій, або регресійний аналіз, включно з передбачуванням часових рядів, наближенням пристосованості та моделюванням.

- UML (англ. *Unified Modeling Language*) – уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю.
- Односторінковий застосунок (англ. *single-page application, SPA*), також відомий як односторінковий інтерфейс (англ. *single-page interface, SPI*) – це веб-застосунок чи веб-сайт, який вміщується на одній сторінці з метою забезпечити користувачу досвід близький до користування настільною програмою.

В односторінковому застосунку весь необхідний код: HTML, JavaScript, та CSS завантажується разом зі сторінкою , або динамічно довантажується за потребою, зазвичай у відповідь на дії користувача. Сторінка не оновлюється і не перенаправляє користувача до іншої сторінки у процесі роботи з нею. Взаємодія з односторінковим застосунком часто включає в себе динамічний зв'язок з веб-сервером.

ВСТУП

Прогнози стали частиною сучасного світу. Економіка складається з прогнозів: політичної обстановки, появи та популярності технологій, врожаю на аграрному секторі, погоди, поточної ситуації з видобутком корисних копалин для різноманітних галузей, та прогнозів вартості різноманітних активів та валют. Всі ці види поточної інформації та передбачень впливають на попит економічного ринку. А цей попит відображається у серцевині економіки – торгах на біржах, які є відображенням капіталістичного устрою людської цивілізації у 21-му столітті.

Торгівля є рушійною силою усього обороту товарів, послуг та грошей.

Тому не дивно, що для того, щоб ефективно працювати на ринку необхідно мати релеванту інформацію, своєчасні прогнози для передбачення майбутнього стану речей, котирування того, чи іншого товару.

І саме прогноз та передбачення є темою даної магістерської дисертації, а конкретно – спеціалізація на прогнозуванні курсу криптовалют.

Криптовалюти були обрані виходячи з наступних положень.

На сьогоднішній день існує понад 2500 криптовалют [1], сім із них мають капіталізацію в мільярди доларів, сотні інших – в сотнях мільйонів доларів.

Користувацькі впровадження різних криптовалют злетіли: мільярди в ринковій капіталізації і мільйони гаманців, за оцінками, були «активними» в 2016 році [2].

Інноваційність, великі перспективи та зацікавленість світової спільноти робить криптовалюти цінним активом.

Як для валют існують спеціальні валютні біржі, так і для криптовалют, які є винаходом недавнього часу, були створені криптовалютні біржі.

В той час, як суть та механізм торгів залишаються такими ж самими – змінюються параметри активів, а саме їх волатильність. На ринку

криптовалют прибутковість від угоди може досягати тисячі відсотків. До того ж є цінною сама технологія, на якій засновані дані цифрові гроші – блокчейн, якій пророчать майбутнє у децентралізації ринку, безпеку та анонімність транзакцій.

Вищенаведена інформація доводить актуальність дослідження у цьому напрямку, тому були поставлені наступні задачі:

- провести аналіз існуючих методів та практик прогнозування курсу криптовалюти;
- розробити метод оптимізуючий процес прогнозування курсу криптовалюти;
- розробити концептуальну модель програмного засобу прогнозування курсу криптовалюти;
- розробити об’єктно-орієнтовану модель програмного засобу прогнозування курсу криптовалюти;
- обрати форму репрезентації програмного засобу прогнозування курсу криптовалюти;
- обрати стек технологій для реалізації програмного засобу прогнозування курсу криптовалюти;
- розробити програмний засіб прогнозування курсу криптовалюти;
- дослідити реалізований програмний засіб прогнозування курсу криптовалюти на ефективність.

Дана робота є описом виконання вищезазначених задач з висновками отриманих результатів.

1. АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ

1.1. Прогнозування курсу криптовалют

Зазвичай прогнозування орієнтоване на отримання прогнозу майбутніх подій. Однак, з точки зору аналітичних технологій прогнозування можна розглядати більш широко. Практично, як завдання прогнозування валютного курсу може розглядатися визначення майбутнього курсу за набором пов'язаних з ним значень: попереднього курсу, активності угод, їх типу, часу та вартості [1-3].

Для прогнозування використовуються формалізовані, евристичні та комплексні методи прогнозування, з використанням різних підходів збору інформації: науковий аналіз, визначення і аналіз причинно-наслідкових та інших зв'язків між ними [4-5].

Для прогнозування валютного курсу використовують методи фундаментального (ФА) і технічного (ТА) аналізу, які є основою інструментарію валютної торгівлі.

Ми розглянемо ТА для нашої мети, так як саме його підхід доцільно автоматизувати та ставити для роботи в режимі реального часу.

Всі методи ТА можна розділити на три групи: графічний, теорія циклів і математико-статистичний. До останньої групи відносять методи, в основі яких лежать формалізовані моделі, що описують закономірності поведінки макроекономічних показників (валютного курсу), побудовані за допомогою відомих математичних і економічних теорій.

Для прогнозування валютного курсу в практичній діяльності трейдера широко застосовуються всі три групи методів технічного аналізу. Загальним недоліком методів можна вказати «ручний» спосіб і суб'єктивізм прийняття рішення в реалізації торгових операцій.

Низька достовірність застосовуваних методів ТА [5], пояснюється тим, що вони припускають тимчасові ряди ВК однорідними і підкоряються одним законом розподілу (нормальному в теорії ефективного ринку).

1.2. Методи прогнозування курсу криптовалют

Наразі кожний трейдер має вбудовані найпоширеніші індикатори та осциляторів, але відомо, що їх застосовність в реальній торгівлі невисока [6].

Тому ми не будемо їх детально розглядати, а почнемо одразу з найпоширеніших методик.

Методика аналізу і прогнозування валютного ринку [6] включає систему генерації сигналів торгової системи (купити, продати, утримувати, закрити поточну позицію). Для генерації та ідентифікації сигналів використовується аналіз перетину декількох експоненційних ковзних середніх з різними періодами усереднення. Критерієм відкриття позицій на купівлю / продаж є емпірично знайдені порогові значення величини середньої кількості всіх поданих сигналів. Критерієм закриття позиції – рівень прибутку / збитку, що задається експертом. Порівняння ефективності методики на різних частотах тимчасового ряду показує, що найбільша ефективність досягається на часових даних. Недоліком є використання ковзних середніх, відомо, що вони часто дають помилкові сигнали, особливо у флеті, запізнювання сигналу під час тренду і розвороту тенденції, що призводить до втрати прибутку. Перевагою методики є простота використання, недоліком – неточність сигналів опису ринкової ситуації, низька ефективність при бічному тренді.

Моделі прогнозування прибутковості валют (і портфеля інвестора) на BP FoRex [7] – побудовані на основі виділення періодичних компонент ряду прибутковості методами сингулярного спектрального аналізу. Аналізується ряд прибутковості у вигляді логарифмічних різниць валютного курсу. Для визначення періодичних компонент і їх характеристик застосовується метод

передбачення головних компонент (PCLP) Д. Тафтса і Р. Кумаресана. Розрахунки по запропонованим моделям відповідно до розроблених методиками є обчислювальною задачею великої місткості. Зазначений недолік обмежує можливості використання даного підходу для активної стратегії трейдера на ринку Forex в швидко мінливих умовах.

Методика побудови ефективних трендових торгових стратегій з використанням індикатора і осциляторів RSI, Stochastic [8] – моделі являють собою рівняння регресії, в яких керуюча змінна (КЗ) представляється лінійною функцією деяких характеристик осциляторів. Значення КЗ інтерпретуються як критерії достовірності сигналів на вхід в ринок (вихід з ринку). У торговельній стратегії для визначення поточної (довгострокової) тенденції використовується технологія «потрійного вибору» і індикатор RSI, тобто ідентифікація поточного тренду здійснюється поза моделлю. Отримано чотири моделі для чотирьох стратегій. Результати апробації моделей на часових даних курсу GBP / USD для всіх чотирьох стратегій показують, що додавання керуючої змінної (моделі) до стандартних стратегій (без моделі) практично не змінює ситуацію.

Методи фрактального аналізу і апарат теорії нечітких множин – використовуються у дослідженні часового ряду валютного курсу пари євро-долар і ряду його збільшень з метою виявлення довготривалої пам'яті, її глибини, трендовіткості, циклів [9]. Для визначення рівня ризику використовується фрактальна характеристика тимчасового ряду – показник Херста.

Результати фазового аналізу часових рядів підтверджують наявність довгострокової пам'яті, яка пояснюється присутністю циклів. Запропонована шестикольорова кусочно-автоматна прогнозна модель, в якій вихідний числовий ряд збільшень валютного курсу перетворюється в лінгвістичний шляхом заміни числових значень термами (лінгвістичними змінними) з урахуванням глибини пам'яті досліджуваного ряду.

Результат прогнозу (величина приросту валютного курсу на день) представляється у вигляді нечіткої множини, який потім трансформується в числовий еквівалент за допомогою процедури дефазифікації. Модель і метод прогнозування призначені для передпрогнозного етапу дослідження ВР. Область застосування отриманих результатів – короткострокове прогнозування критичних тенденцій на валютному ринку в якості додаткового інструменту трейдера.

Методика факторно-регресійного аналізу – відслідковує вплив агрегованих факторів на номінальні і «інваріантно-індексні» курси валют, за допомогою чого розроблені нові економетричні моделі регресії мінової цінності валюти на виділену групу агрегованих факторів. Також запропоноване використання у дослідженнях економіко-математичних моделей впливу макроекономічних факторів на валютні курси, в основу яких покладено поняття індексу мінової цінності валюти [10]. Показники інваріантних індексів мінової цінності валюти (адитивний і мультиплікативний) використані для модифікації стандартних (структурних) моделей для визначення рівноважного валютного курсу.

У рамках методики було розглянуто модель Манделла-Флемінга, монетарні моделі з гнучкими і з жорсткими цінами Дорнбуша та побудована регресія індексу мінової цінності британського фунта на виділені 4 головні компоненти.

Але низькі значення коефіцієнта детермінації рівняння регресії – 0,309 і величин питомого впливу кожного агрегованого фактора на валютний індекс британського фунта свідчать про невисоку практичну значущість отриманої моделі для прогнозування валютного курсу.

Можливою причиною є «змішана» (неоднорідна) природу валютного курсу і можливості розбиття тимчасової кривої на «фрактальні інтервали» [11]. Як показник неоднорідності кривої тимчасового ряду валютного курсу

використовується фрактальна розмірність, метод визначення якої, полягає у вимірюванні довжини кривої в різних масштабах часу.

Виділено три інтервала змін значень фрактальної розмірності: персистентний (відповідає тренду), стохастичний (відповідає флет) і антиперсистентний інтервал (попереджає про зміну тренду). Недоліками є: суб'єктивний вибір «характерних ділянок» тимчасової кривої для розрахунку фрактальної розмірності, предпрогнозний характер методу і низький відсоток точності прогнозу на двох останніх інтервалах.

Методика з використанням APIЗС – моделі авто-регресії інтегрування змінного середнього широко використовуються при аналізі часових рядів [12].

Переваги:

- чітке математико-статистичне обґрунтування;
- формалізована і найбільш докладно розроблена методика, за якою можна підібрати модель, найбільш придатну для до кожного конкретного тимчасового ряду, а точкові і інтервальні прогнози випливають з самої моделі і не вимагають окремого оцінювання;
- ефективні прогнози на малі часові проміжки.

Недоліки:

- для побудови потрібно не менше 40 спостережень, а для SARIMA – близько 6-10 сезонів [12];
- неадаптивність – при отриманні нових даних, модель потрібно періодично переоцінювати, а іноді переідентифіковувати;
- побудова задовільної моделі вимагає великих витрат ресурсів і часу.

Доцільно використовувати при короткотривалому прогнозуванні, може використовуватися на середньотривалому і недоцільна на довготривалому прогнозах.

Методика експоненціального згладжування – метод, який обчислює значення згладженого ряду шляхом оновлення значень, визначених на

попередньому кроці на основі інформації з поточного кроку [13]. Інформація з попереднього і поточного кроків береться з різними вагами, якими можна керувати. Простий, ефективний метод, але недоцільний при короткотривалому прогнозуванні і зовсім не ефективний на довготривалому.

Переваги методу експоненційного згладжування:

- ефективні прогнози на середньострокові терміни;
- адаптивність до надходження нових даних для обробки;
- простота алгоритму та використання;
- проміжні результати доступні для користувача.

До недоліків методу експоненційного згладжування можна віднести малу точність прогнозів для малих та великих часових проміжків.

Марковські випадкові процеси та Ланцюги Маркова – частковий випадок випадкових процесів, що широко застосовуються для створення стохастичних (імовірних) моделей, котрі описують процеси функціонування певних систем технічного, економічного, екологічного та соціального профілю, центральне місце належить марковським.

Випадковий процес $X(t)$ називають марковським [14], якщо за будь-якого можливого значення часу $t = t_1$ значення випадкової величини $x(t_1)$ не залежить від того, яких значень ця величина набувала для $t < t_1$, тобто процес у момент часу $t = t_1$ не залежить від його поведінки в більш ранні моменти часу $t < t_1$.

Процеси зміни курсів обміну є процесами стохастичними, тобто представляють множину як визначених (тенденційних), так і випадкових подій. Звідси випливає висновок про доцільність застосування перехідних ймовірностей Маркова, які характеризують процес переходів (наприклад, у часі) випадкової величини χ з однієї умови χ_i в іншому умові χ_j .

Якщо говорити про процес зміни котирувань валют, то перехідні ймовірності – це умовні ймовірності $p_{i,j}$ того, що в момент часу t поточне значення обмінного курсу j за умови, що в момент $t - 1$ він дорівнював i .

Вищезазначені процеси можна побачити на рис. 1.1.

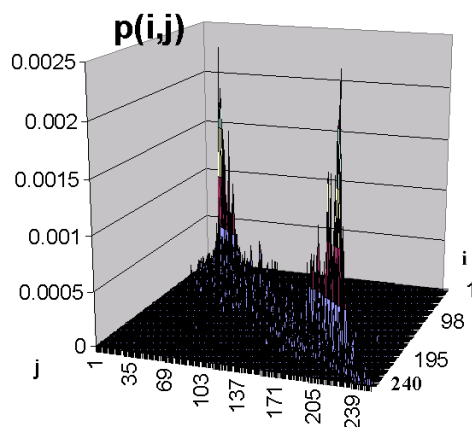


Рис. 1.1. Розподіли імовірностей $P(i, j)$ Маркова

Основною властивістю ймовірностей Маркова є пам'ять про попередні переходи. Ця властивість в контексті розглянутої проблеми може бути сформульована таким чином: розподіл ймовірностей $P_t(i, j)$ характеризує ймовірність того, що котирування валюти прийме значення j , за умови, що після t кроків (наприклад, t годин) котировка рівна i [15].

Дані алгоритми реалізовані в декількох програмних продуктів у якості основних та допоміжних засобів для прогнозу, та мають гарні результати на коротких та середніх інтервалах.

Методика використання штучних нейронних мереж – застосовує нейронну мережу, яка є системою простих процесорів (нейронів), що сполучені і взаємодіють між собою. Для прогнозування є два типи нейромереж: перцептрон та мережі з архітектурою Радіально Базисної Функції (РБФ) [16]. Гарні результати на всіх строках прогнозування, через відстеження та копіювання шаблонів поведінки людей. Адже саме люди та їх активність впливає на тренди та торги вцілому. Хоча існує достатньо математичних моделей, які можуть прогнозувати ціни по попереднім числовим значенням – всі вони недосконалі, через неможливість повністю описати математично поведінку людей, як гравців.

Нейромережі кращі в цьому саме через виявлення шаблонів «людських дій» через числові значення, це дає можливість давати гарні прогнози на будь-яких строках [17].

Переваги та недоліки прогнозування за допомогою ШНМ описані у табл. 1.2.

Таблиця 1.2

Переваги та недоліки прогнозування з використанням ШНМ

Переваги	Недоліки
Ефективність прогнозів на малі, середні та помірно великі часові проміжки	Складність аналізу на пошук проблеми при помилках прогнозування
Адаптивність до надходження нових даних	Робота за принципом «чорного ящика», що не дає доступу до проміжних результатів складання прогнозу
Можливість передбачення динамічних процесів	Доволі складний процес емпіричного «навчання» та перевірки результатів
Має можливість навчатись за допомогою декількох навчальних шарів, якими можуть слугувати іншу методи ТА	Можливі виліти в самозбудження (коли сигнал, отриманий з виходу, забиває все, що приходить по входах)

Аналізуючи всі перераховані праці можна побачити, що в основному запропоновано прогнозування майбутнього значення валютного курсу методами:

- множинної регресії;
- сигнального методу з математичним апаратом на основі спектрального і фрактального аналізів;
- формалізації керуючої змінної на основі сигналів відомих індикаторів та осциляторів;
- товарним та монетарним підходом.

З представлених даних ми бачимо, що наявні методи технічного аналізу не дозволяють побудувати ефективні торгові прогнози для реальної торгівлі на валютному ринку на всіх групах діапазонів [18], та часто є недієздатними для роботи в режимі реального часу через велику місткість задачі математичного апарату, що лежать в основі методів.

Як результат ми маємо велику кількість осциляторів, індексів, допоміжних величин, навіть вбудованих в біржі [19]. Тобто дані технічного аналізу, які допомагають при торгівлі. Але для фінального рішення гравці керуються методами формального аналізу у поєднанні з підказками технічного.

Поточна ситуація робить торгівлю для неосвічених гравців навіть на звичайному валютному ринку занадто ризиковою, вже не кажучи про криптовалютний ринок.

1.3. Програмні засоби прогнозування курсу криптовалют

Trader – система дозволяє прогнозувати рух курсів валют на валютній біржі. Як вихідна використовується інформація за попередніми результатами торгів (часовий ряд): максимальна, мінімальна ціна, ціна закриття і обсяг угод за день. В системі застосовуються наступні алгоритми для аналізу даних:

- ковзне середнє трьох видів:
 - лінійне;
 - експоненціальне;
 - з вагами, що задаються.
- MACD-гістограми;
- Індикатори:
 - RSI;
 - OBV;
 - Williams R%;

- CandleSticks;
- Point & Figure.

Користувач може створювати власні формули для аналізу даних. До переваг також можна віднести можливість застосування індикатора до вже побудованого індикатора, що наприклад потрібно при побудові MACD-гістограми, де ковзне середнє обчислюється для різниці двох ковзних середніх.

Серед недоліків виокремлюється тільки неоптимальна зручність самого продукту для користувача.

Walletinvestor – має вигляд веб-додатку з можливістю вибору однієї конкретної криптовалюти, або ж спостерігати дані про всі наявні з них у вигляді таблиці. Має прогнози на періоди:

- два тижні;
- три місяці;
- пів року;
- один рік;
- п'ять років.

Програму можна відвідати за посиланням <https://walletinvestor.com/forecast>. Сайт пропонує користувачу окрім прогнозів поточний курс найпопулярніших криптовалют і усю попередню інформацію по , посилання на біржі та прогнози від інших компаній, а також дає змогу безпосередньо придбати криптовалюту за посиланням на їх сайт-відгалудження. Методи, які використовуються для прогнозів не розголошуються з причини комерційної таємниці. Сторінки та робота веб-додатку продемонстровані на рис. 1.2 та рис. 1.3.

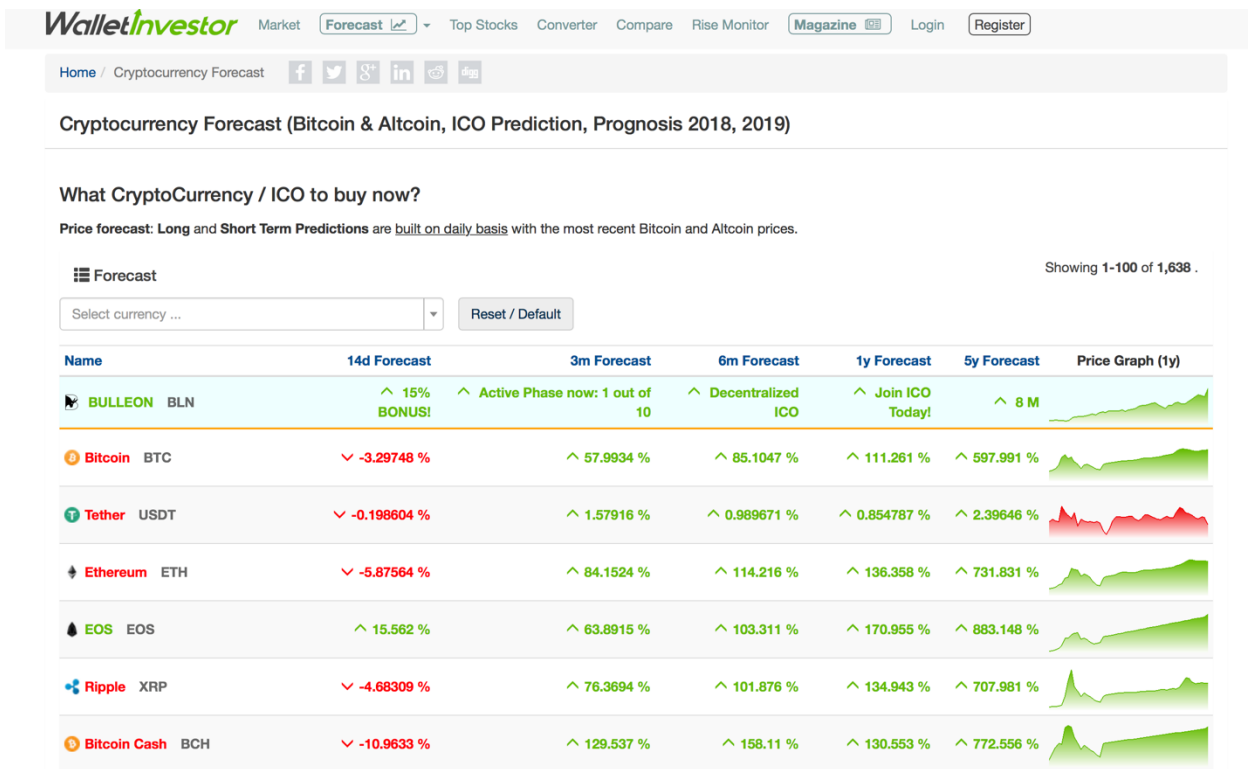


Рис. 1.2. Робота з веб-додатком для прогнозів Walletinvestor

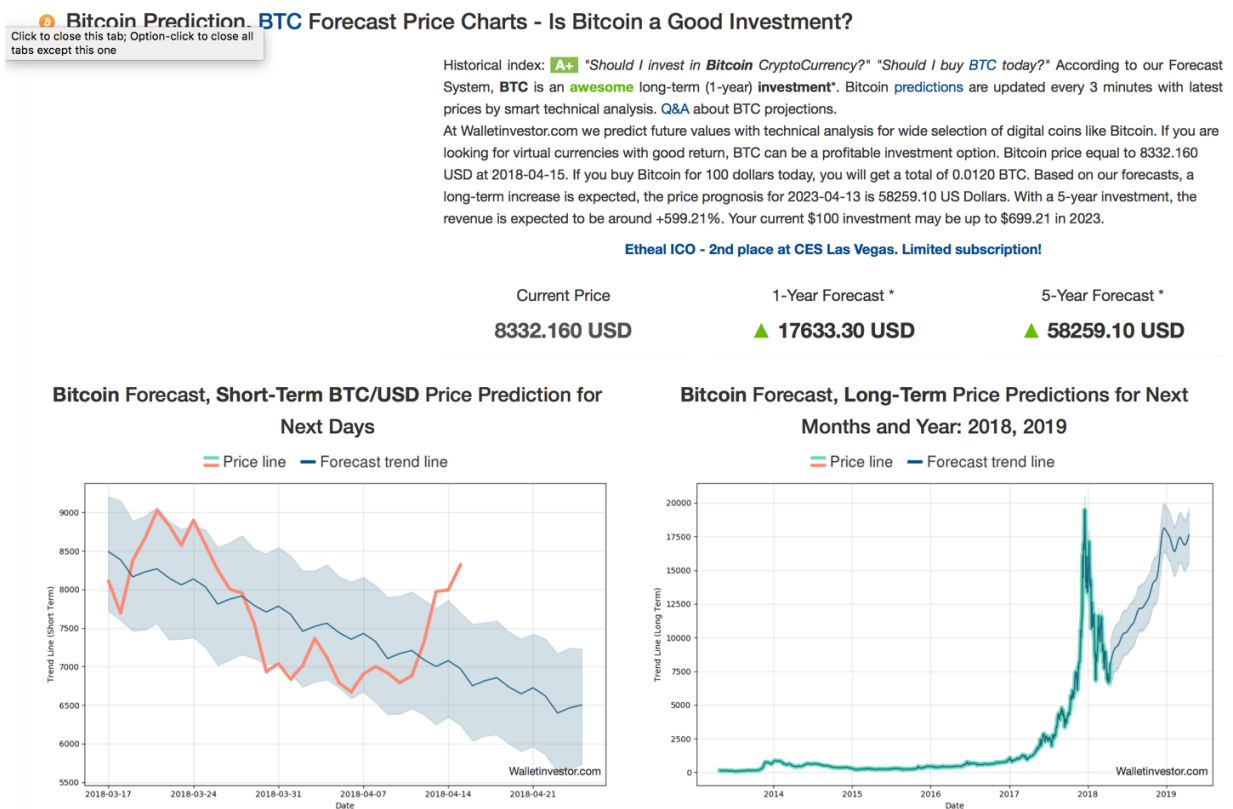


Рис. 1.3. Робота з веб-додатком для прогнозів у деталях про Bitcoin

Belinvestor – веб-додаток, який розташований за адресою <https://belinvestor.com/cryptocurrencies/>. Пропонує прогнози на найближчі дати у форматі стрічки новин. Вигляд сайту продемонстровано на рис. 1.4.

Основною проблемою є те, що на сайті пости з давністю більше трьох місяці – видаляються, що не дає змоги подивитися старі прогнози, та оцінити їх точність. По новим прогнозам можна сказати, що їх складають люди з досвідом та знанням сфери, але їх ефективність мала з приводу очевидності надаваних трендів.

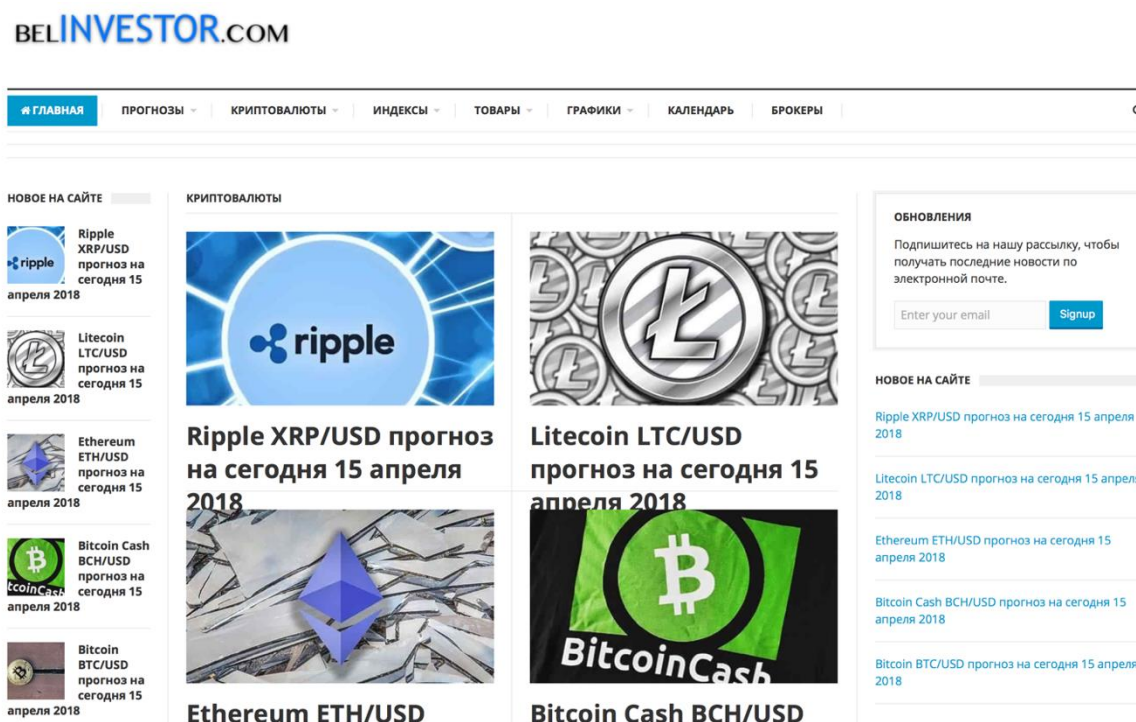


Рис. 1.4. Робота з веб-додатком для прогнозів Belinvestor

Кожний прогноз має вигляд окремої статті з історією криптовалюти. Також присутній опис, чинники, які можуть впливати на її курс.

Головим чином там описано хід думок аналітика, який складав даний прогноз. Це показує, що основним джерелом прогнозів є фундаментальний аналіз та команда професіоналів, які стоять за ними.

На сайті не розголошують методи, якими користуються експерти для встановлення прогнозів.

Вигляд статті з прогнозом можна побачити на рис. 1.5.



Рис. 1.5. Стаття з сайту Belinvestor з прогнозами на курс криптовалюти Bitcoin

NeuroShell – являється цілим набором нейронних мереж, які створені та навчені з єдиною метою – передбачення курсу котирувань валют на фінансових ринках. Оскільки сам принцип застосування ШНМ полягає в процесі схованим у «чорному ящику» – користувач не повинен алгоритму обробки, чи певних його частин. А за допомогою мінімалістичного інтерфейсу все приходить лише до задання параметрів та отриманню результату. Тобто з цією процедурою може впоратись людина з будь-яким

рінем досвіду у фінансових операціях. *NeuroShell Day Trader* має ще одну сильну сторону – він вміщує в своїх методах обробки оптимізації, засновані на принципах генетичних алгоритмів. Їх впровадження змінило підхід для підготовки ШНМ у контексті вибору імпірично кращих параметрів для індикаторів та обробку даних з різних входів робочих шарів мережі.

NeuroShell Day Trader концентрується на побудові торгової системи. Сама ж торгова система може використовувати в своїх правилах як індикатори, так і спрогнозовані значення, отримані за допомогою нейронних мереж. Сам процес побудови нейронних мереж досить простий, однак жоден з ключових етапів побудови нейронної мережі залишається в тіні. Основна архітектура, яка використовується в *NeuroShell Day Trader* – багат шаровий персептрон.

Elliott wave analyser professional 6.0 – програмний продукт призначений для аналізу валютного ринку з використанням принципів теорії хвиль Елліотта (ТХЕ), а також за допомогою стандартних алгоритмів технічного аналізу. У 1930 році Ральф Елліотт виявив, що емоційний стан натовпу впливає на курси валют, і цей вплив описується декількома зразками, які тепер відомі як хвилі Елліотта.

Програмний продукт дозволяє виділяти з тимчасового ряду незакінчені зразки, які можна віднести (певною мірою, звичайно) до зразків хвиль Елліотта. Оскільки поведінка стандартних хвиль Елліотта вивчена, то можна зробити прогноз щодо подальшого розвитку цих незакінчених зразків. Для кожного із зразків (закінчених і незакінчених) система обчислює коефіцієнт Goodness, змінюваний від 0 до 100, який і визначає ступінь близькості досліджуваного зразка до теоретичного аналогу.

Одним з важливих параметрів аналізу є кількість міток – число відрізків ламаної, яка апроксимує досліджуваний зразок. Користувач може задати щільність розподілу міток. Як правило, до кінця досліджуваного проміжку часу слід збільшити щільність розподілу міток.

Після робиться аналіз кожної з отриманих хвиль (тип, розмір, завершеність) і даються сигнали входу або виходу з ринку для коротких або довгих позицій.

Дані про курси можна об'єднувати в групи. Тоді після аналізу валюти, за якими були згенеровані сигнали входу і / або виходу, виділяються кольором (червоний – вихід, синій – вхід). Користувач може налаштовувати параметри, за якими визначаються критерії виходу і входу.

У разі роботи в режимі on-line, програма автоматично виробляє перерахунок хвиль, період якого задає користувач.

Поряд з аналізом за методом Елліотта, користувач може сам написати будь-який алгоритм індикатора на будь-якій мові програмування, і обчислений індикатор висвічуються на тому ж графіку, що і вихідні дані. На один і той же графік можна накласти багато індикаторів, що дозволяє одночасно бачити їх показання.

Користувач може задати діапазон аналізованих даних по датах, а після аналізу порівняти реальні а передбачені результати.

В системі є докладна довідкова система з теоретичним описом принципу Елліотта, а також розділ Guided Tour, який проводить користувача через всі етапи аналізу даних по методу Елліотта. На жаль демо-версія програми аналізує тільки свої власні приклади.

ТХЕ як інструмент дозволяє отримати довгострокові прогнози, не такий ефективний короткострокових та середньострокових прогнозах через слабку кореляцію трендів з намірами гравців.

Ainet – для прогнозування подій використовуються нейронні мережі. Для аналізування часових рядів дана програма мало придатна, але дає хороші результати для багатьох завдань з інтерполювання даних.

Параметр аналізу – *penalty coefficient*, який програма сама ж і оптимізує. В якості вихідних даних виступає прямокутна матриця з повністю присутніми даними і матриця з тим же кількістю стовпців, але в якій має

місце відсутність деяких даних. Програма намагається передбачити значення цих відсутніх даних.

Серед недоліків програми виокремлюється те, що вона працює за методом «чорної скриньки», Це призводить до складнощів розуміння користувачем нейронних мереж, що використовуються для побудови прогнозів.

1.4. Висновки

За результатами порівняння програмних засобів прогнозування курсу криптовалют були виділені їх переваги та недоліки. З огляду на це виокремлено Ainet, як засіб з кращими відгуками та результатами прогнозування на основі нейронних мереж.

Проаналізовано методи прогнозування курсу криптовалют з основами на математичному підґрунті і формулах як:

- теорія хвиль Елліота;
- метод експоненціального згладжування;
- Марковські випадкові процеси та ланцюги;
- АРІЗС.

А також прогнозування на основі Марковських випадкових процесів та ланцюгів і ШНМ на архітектурі багатошарового перцептрона, які були визначені як найефективніші.

2. МЕТОД ТА СТРУКТУРНА МОДЕЛЬ ПРОГРАМНОГО ЗАСОБУ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ

2.1. Постановка задачі

Для того, щоб розробити повноцінний програмний засіб для прогнозування курсу криптовалют поставимо наступні вимоги:

- запропонувати новий метод прогнозування курсу криптовалют;
- формалізувати діяльність з прогнозування курсу криптовалют;
- встановити зв'язки між етапами прогнозування курсу криптовалют;
- встановити потоки даних між етапами прогнозування курсу криптовалют.

2.2. Новий метод прогнозування курсу криптовалют

За висновками попереднього розділу видно, що існуючим методам не дістає точності для надання остаточних прогнозів, якими одразу можна було користуватися з мінімальним ризиком. Тобто таким ризиком, який відповідає точності прогнозу більш ніж 50 відсотків.

Окрім цього не всі більш-менш ефективні методи оптимізовані саме під ринок криптовалют, який має свої незначні відмінності у шаблонах торгівлі та волатильності активів.

Отже метою є:

- розробка методу прогнозування курсу валют з відсотком точності прогнозів більше 50%;
- адаптація методу під ринок криптовалют.

Розумно буде обрати за основу для оптимізації виділені найефективніші методи: прогнозування на основі Марковських випадкових процесів [20] та ланцюгів і ШНМ з архітектурою багатоварового перцептронну.

Але для створення єдиного методу слід органічно поєднати дані методики для отримання єдиного більш точного прогнозу. І саме тут ШНМ також надають ідеальне рішення – налаштувати результати методик як вхідні дані ШНМ на основі багат шарового перцептрона навченої по алгоритму зворотнього поширення помилки або скорочено – багат шарового перцептрона Румельхарта.

Багат шаровий перцептрон Румельхарта – є розширеною та структурно зміненою версією перцептрона Розенблатта [21]. Відмінність полягає в тому, що алгоритм зворотнього поширення помилкового сигналу передається для навчання всім шарам. Окремо зазначається присутність більше єдиного навчального рівня. Стандартом в даному випадку є використання двох чи трьох шарів, так як експериментально не доведено ефективність наявності більшої кількості шарів, навпаки нейронна мережа стає повільнішою, а точність прогнозів не збільшується.

Отже ми маємо рішення для об'єднання декількох методик, та рекомендоване число шарів: в нашому випадку це два навчальних шари.

Слід відмітити, що результуюча ШНМ має отримувати у якості вхідних даних не параметри валюти на ринку, а вже отримані прогнози по цим параметрам від інших методів, по яким вже за допомогою навчання нейромережі з двома схованими шарами та навчанні зі зворотнім поширенням помилки буде видано найбільш вірогідний результат прогнозу, який повинен нівелювати слабкі сторони кожного з попередніх методів, та утвердити їх сильні сторони.

Інакше кажучи: після обробки декількома методами ми будемо мати часовий ряд курсу, або ж єдине значення тренду від кожного.

На наступному етапі результуюча ШНМ навчаючись на даних з часових рядів буде згладжувати, або ж повністю виправляти місця в ряді, де один з методів проявив себе гірше іншого – збільшуючи ваги правильного методу, та зменшуючи ваги методу, що помилився.

Це надасть значне покращення результатів виключаючи можливість високих діапазонів розбіжності прогнозу з реальністю.

А на рівні, де обидва методи мають високу точність прогнозу, в залежності від кількості вибірок для навчання, результат може бути згладжений, чи усереднений, що можливо вийде нам у менший відсоток вищого порогу точності на вже існуючих даних, але збільшить цей відсоток при отриманні нових даних у режимі реального часу.

Як результат, метод застосовує дві окремих ШНМ на різних етапах обробки даних, що є ресурсозатратним, як при моделюванні та реалізації методу, враховуючи необхідне навчання обох мереж на підготовлених вибірках даних, так і при обслуговуванні розробленого програмного засобу.

З цих причин для попередньої обробки даних було обрано лише два методи, щоб зменшити складність та ресурсоемкість.

Але в подальшому ми маємо потенціал для розширення.

Підсумовуючи вищезазначене ми маємо алгоритм нового методу для прогнозування курсу криптовалют:

1) Збір даних про криптовалюту, які включають в себе:

- найменування та ідентифікаційний номер криптовалюти;
- поточний курс, та часовий ряд курсу у минулому;
- часовий ряд кількості та типу (продаж / купівля) укладених угод;
- часовий ряд загальної суми угод у одиницях криптовалюти;
- часовий ряд максимумів та мінімумів об'єму угод.

2) Подача набору даних паралельно до двох методів для обробки:

- методом випадкових процесів та ланцюгів Маркова;
- методом обробки ШНМ на архітектурі багатоварового перцептрону навченої по алгоритму зворотнього поширення помилки.

- 3) Передача отриманих часових рядів прогнозу на обробку ШНМ на архітектурі багатoshарового перцептрону навченої по алгоритму зворотнього поширення помилки на вибірках прогнозів.
- 4) Видача фінального часового ряду прогнозу.

2.3. Структурна модель програмного засобу

Для забезпечення вищезазначених вимог та опису створеного методу було вирішено обрати моделі, показані на рис. 2.1, як опис основних механізмів роботи програмного засобу.

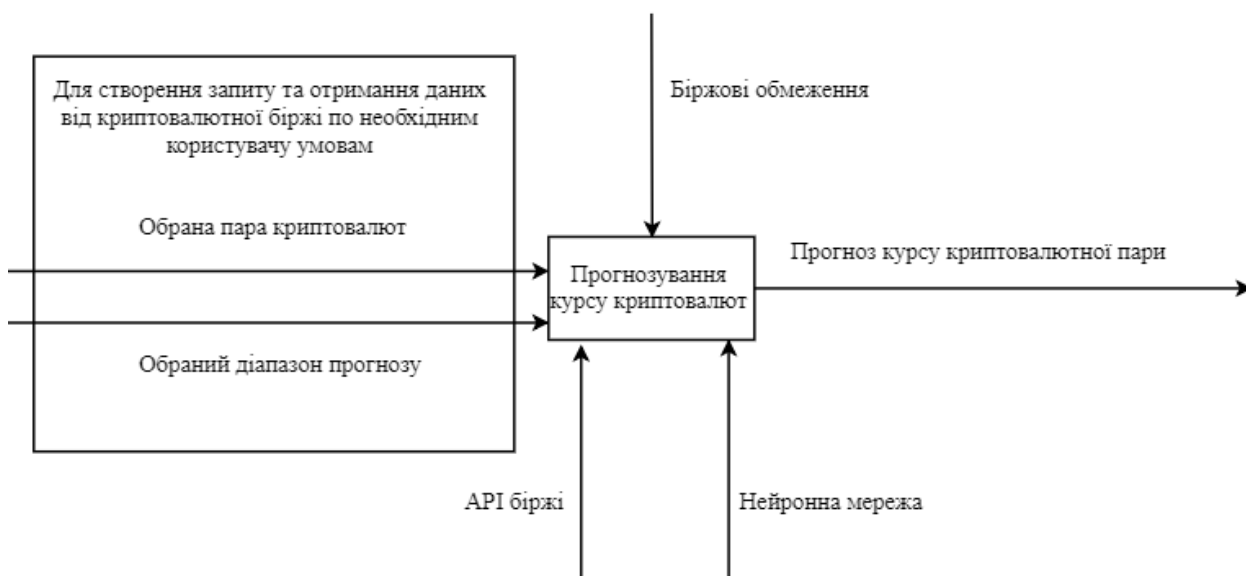


Рис. 2.1. Концептуальна модель програмного засобу

Концептуальна модель має такі дані:

Вхідні дані: «Обрана пара криптовалют» і «Обраний діапазон прогнозу» на основі яких створюються HTTP-запит до API біржі, та обираються параметри при обробці нейромережею архітектури рекурентного перцептрону [4] та математичними алгоритмами на основі Марковських випадкових процесів та ланцюгів, методу експоненціального згладжування, індикатора РСС і осциляторів R [22] з подальшим оновленням даних.

Також слід зазначити, що найчастіше ціна валюти визначається її співвідношенням до доллару (USD) або біткойну (BTC), але для більшої гнучкості програмного засобу було вирішено надавати користувачу можливість самостійного вирішення у співвідношенні до якої криптовалюти вираховувати курс.

Через це у криптовалютній біржі запитуються дані по обом обраним валютам, що дещо сказується на часі обробки, але не сильно, адже глобальні дані історії кожної криптовалюти вже були запитані у бірж, ще на етапі навчання ШНМ, а в ході реальної роботи програми обсяг та минулий часовий період запитуємих даних обмежений:

- якщо необхідний короткостроковий прогноз – запитуються дані найближчого часу з найменшими часовими інтервалами (1-30хв);
- якщо необхідний довгостроковий прогноз – запитуються дані за великий попередній період, згруповані по значним часовим проміжкам (1-7 днів).

Регулюючі чинники: біржові обмеження по можливій частоті та кількості запитів [23]. А також в залежності від обраної біржі може бути доступним різний діапазон попередніх даних, та їх різний набір, тобто різні біржі по-різному визначають якими даними ділитися, а якими ні, судячи з їх власних міркувань.

Можливі біржі для роботи: Binance, KUNA, YoBit [9].

Механізми роботи поділені на дві основні частини:

- *API біржі* – надає запитані данні відносно обраної пари криптовалюти у заданому часовому діапазоні, на кшталт [24] значення курсу, кількості угод, їх мінімуму та максимуму, загальній сумі продаж, часу дня, та активності, тощо.
- *Штучна нейронна мережа* – є сукупністю декількох нейромереж і є не просто програмною частиною, але самостійним модулем, так як перед реальним використанням кожен з них необхідно навчити на

спеціальних вибірках даних [4, 6], після чого вони підключаються до програми в якості засобу обробки біржових даних для видачі прогнозу, а також обробки всіх прогнозів для зіставлення і надання найвірогіднішого.

Вихідні дані: прогноз курсу криптовалютної пари, який надається у вигляді часового ряду на заданий вхідними даними інтервал, а графічним інтерфейсом програми відображається у вигляді таблиці чи графіку.

2.4. Декомпозиція концептуальної моделі програмного засобу в нотації IDEF0

Головні процеси, які виконуються при роботі програмного засобу можна побачити на рис. 2.2.

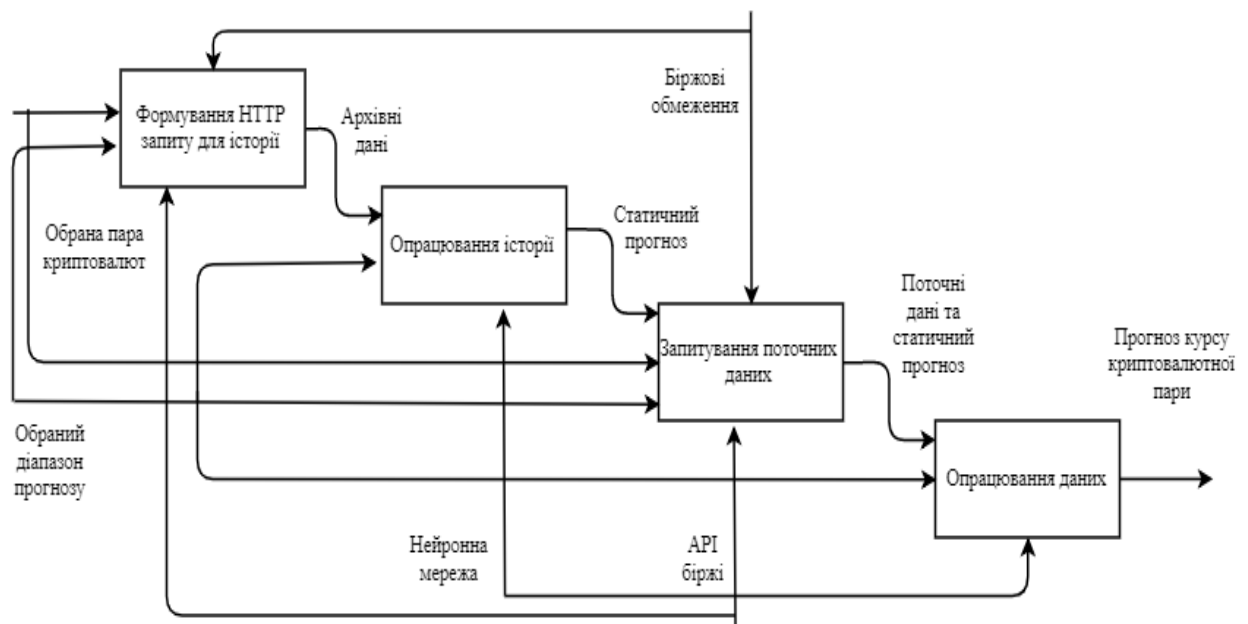


Рис. 2.2. Декомпозиція концептуальної моделі програмного засобу в нотації IDEF0

Запит історії – у цьому процесі створюється HTTP-запит на основі наданих даних про пару криптовалют та часовий діапазон прогнозу, після чого відправляється до API біржі.

Обробка історії – у цьому процесі отримана JSON-відповідь від API біржі парситься (з JSON об'єкту дістаються необхідні для роботи програми дані), з отриманих даних створюються програмні об'єкти для нейронної мережі та математичних алгоритмів описаних вище, вони відправляються на опрацювання після чого всі отримані прогнози надходять на вхід нейромережі для зіставлення прогнозів, яка надасть остаточний результат статичного прогнозу.

Запит поточних даних – процес, який є початком основного циклу програми для підтримки видачі прогнозу в режимі реального часу – тобто режим програми, в якому дані автоматично оновлюються з певним інтервалом та виведенням інформації користувачу.

Даний режим буде працювати на малих інтервалах: від 1 хвилини до 24 годин. Його основа це повторне складання HTTP-запиту і роботу з API біржі. В режимі одиничного інтервалу діапазону, використовуються ті ж дані обраної пари та діапазону прогнозу.

Обробка даних – процес, в якому в залежності від обраного часового діапазону прогнозу до поточних об'єктів даних буде додані нові, після чого заново пройде обробка нейронною мережею та математичними алгоритмами і видано оновлений прогноз, або через несуттєвість часового проміжку отриманих даних в порівнянні з обраним часовим діапазоном прогнозу – дані будуть відсічені: тобто не прийняті до роботи, через їх фактичну ідентичність попереднім. Прогноз в даному випадку залишиться статичним.

2.5. Декомпозиція концептуальної моделі програмного засобу в нотації IDEF3

Основні етапи роботи моделі: Формування запити, Отримання даних, Обробка даних, Видача результату. Вони відбуваються з даними, тому доцільно розглянути декомпозицію у нотації IDEF3, яку можна побачити на рис. 2.3.

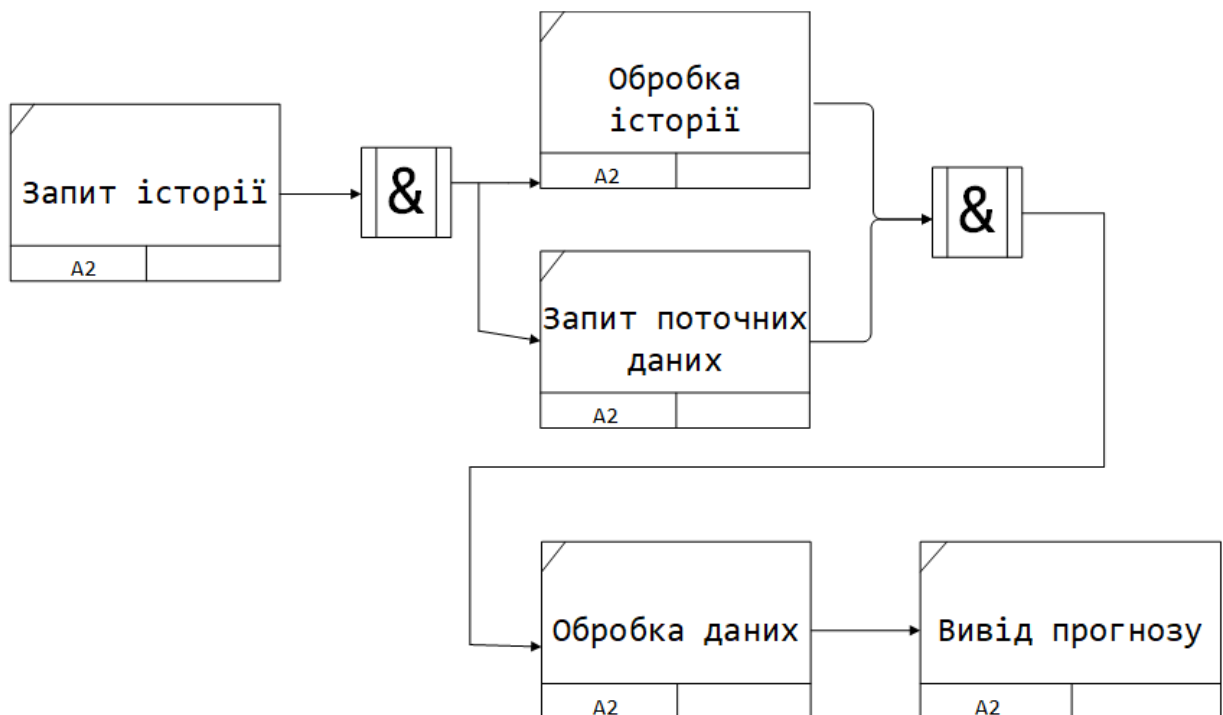


Рис. 2.3. Декомпозиція концептуальної моделі програмного засобу в нотації IDEF3

Процес «Обробка історії» починається з конвертації отриманих від серверу даних у форматі JSON-об'єкту до формату колекції програмних об'єктів, яка потім паралельно передається до модуля з математичними алгоритмами та модуля нейронної мережі. В цих модулях дані незалежно оброблюються та додаються в загальну колекцію прогнозів у вигляді часових рядів. Дана колекція подається в спеціальну нейромережу для обробки прогнозів, яка зіставить їх, та для кожного часового проміжку обере найвірогідніше значення курсу.

Процес «Обробка даних», виконує зіставлення часового діапазону отриманих даних від серверу з обраним в програмі часовим інтервалом. Якщо обраний суттєво великий інтервал, наприклад прогноз на місяць вперед: тоді зроблений статичний прогноз на етапі обробки історії буде зроблений на основі даних якнайменше місячної давності. А отже нові дані в інтервалі хвилин, або годин несуттєві для такого прогнозу. Вони будуть відсікатися до тих пір, поки об'єм нових даних не досягне достатнього обсягу для додання його в часовий ряд програмних об'єктів.

При малому часовому проміжку дані будуть оновлятися часто.

Поточне значення буде конвертоване і додане до колекції, яка буде заново передана до модулів обробки значень, після чого основний прогноз буде оновлено і виведено користувачу у обраному ним вигляді.

Слід виокремити, що на етапі обробки історії і на етапі повної обробки: запит, математична обробка, та обробка нейронною мережею – загалом виконуються єдиний процедурний ряд. Але обробка історії має ініціалізуючий характер, а етап обробки даних виконує головний програмний цикл з умовою для перевірки доцільності обробки поточних даних.

2.6. Декомпозиція концептуальної моделі програмного засобу в нотації DFD

Чітке визначення рамок, процесів і потоків даних в системі можна побачити на рис. 2.4 у декомпозиції концептуальної моделі в нотації DFD.

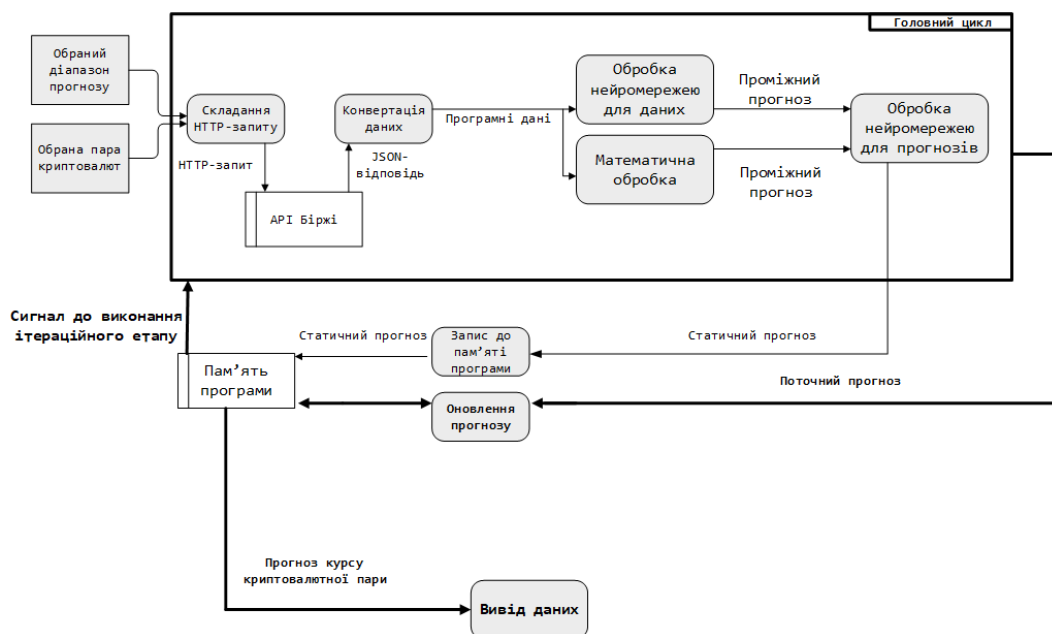


Рис. 2.4. Декомпозиція концептуальної моделі програмного засобу в нотації DFD

Центральним процесом є «*Прогнозування курсу криптовалют*». На вхід поступають обраний діапазон прогнозу та обрана пара криптовалют. Виходом з цього процесу буде часовий ряд з прогнозом курсу криптовалюти, який буде виведений користувачу зручним йому способом: графічне відображення таблицею та діаграмою, текстовий файл, або мінімалістичне повідомлення з контрольними прогнозами.

Сховищами даних виступають :

- *API біржі* – основні джерела даних, які надають доступ по запиті до статистики поточних та минулих угод, їх часу, курсу криптовалют.
- *Пам'ять програми* – зберігає лише одну порцію конвертованих даних та поточних прогноз, отриманий від їх обробки, після нового запиту – конвертовані дані оновлюються, після чого оновлюється прогноз.

Пам'ять програми є проміжним сховищем даних у рамках роботи з користувачем, а після закінчення роботи – всі дані видаляються. Для зрозумілого розділу між фазою ініціалізації програми з основним

заповненням колекції об'єктів, та послідовним оновленням колекції за умовою збігу часового діапазону – було зручно виділити її окремо.

Також на діаграмі зображено дочірні процеси, які відображають потоки даних між ними, а саме:

- «*Складання HTTP-запиту*» – відповідно до заданих користувачем параметрів складається HTTP-рядок, який посиляється до серверу біржі через її API.
- «*Конвертація даних*» – отримані дані в текстовому чи бінарному форматі переводяться в об'єкти у динамічній пам'яті програми для подальшої роботи з ними.
- «*Обробка нейронною мережею для даних*» – дані у вигляді об'єктів віддаються нейронній мережі на основі радіально-базисних функцій для складання атомарного прогнозу [4, 6].
- «*Математична обробка*» – ті ж самі дані паралельно віддаються на входи до вищезазначених математичних алгоритмів [7].
- «*Обробка нейромережею для прогнозів*» – оброблений часовий ряд з прогнозом з обох джерел віддається на вхід до нейромережі на основі рекурентного перцептрону [4, 6], яка конвертує декілька прогнозів в єдиний прогноз.
- «*Оновлення прогнозу*» – доповнюючий запис поточного прогнозу до пам'яті програми у вигляді об'єктів.
- «*Вивід даних*» – вивід даних користувачу у заданому їм форматі.

2.7. Висновки

Формалізований процес прогнозування курсу криптовалют.

Для виконання поставленої задачі відповідно до поставлених вимог встановлені зв'язки та потоки даних між етапами прогнозування курсу

криптовалют, здійснено побудову концептуальної моделі та її декомпозиція у нотаціях IDEF0, IDEF3 та DFD.

Отримані моделі дають чітку основу та розуміння процесів для побудови об'єктно орієнтованої моделі з подальшою реалізацією програмного засобу.

3. ОБ'ЄКТНО-ОРІЄНТОВАНА МОДЕЛЬ ПРОГРАМНОГО ЗАСОБУ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ

Попередньо визначена необхідність прогнозування курсу криптовалют за допомогою відповідного програмного засобу. Його розробленню передують визначення:

- вимог;
- можливостей;
- структури.

3.1. Постановка вимог

Основні функціональні вимоги, що ставляться до серверної та клієнтської частин системи, наведено в табл. 3.1 та 3.2.

Для визначення коректності вимог в цілому для системи необхідно встановити чи не суперечать вони одна одній та чи не перекриваються між собою. Матриці залежності вимог до програмного засобу прогнозування курсу криптовалют наведені в табл. 3.3 та 3.4.

Таблиця 3.1

Функціональні вимоги серверної частини, що ставляться до системи

Кодовий номер	Вимога
B1	Можливість отримання форми з парою криптовалют та часовим проміжком
B2	Можливість створення запиту у форматі HTTP для API принаймні однієї криптовалютної біржі
B3	Можливість прийому відповіді на запит у форматі JSON
B4	Можливість парсингу даних з JSON формату в об'єкти
B5	Можливість запису даних у вигляді об'єктів до бази даних у разі великої кількості запитів до конкретної криптовалюти

Продовження табл. 3.1

B6	Можливість перенаправлення запиту до бази даних, замість криптовалютної біржі, у разі наявності в базі потрібних даних
B7	Можливість передачі даних у вигляді часового ряду об'єктів до класів з методами обробки даних у прогнози
B8	Можливість видачі класами-обробниками вихідних даних прогнозів у єдиному вигляді, по екземпляру з кожного обробника
B9	Можливість обробки екземплярів прогнозів ШНМ для генерації єдиного прогнозу
B10	Можливість запису прогнозу у базу даних, у разі його довгостроковості
B11	Можливість перенаправлення запиту прогнозу до бази даних, для його безпосередньої видачі, у разі збігу параметрів криптовалютної пари та часового діапазону з наявним в базі прогнозом

Таблиця 3.2

Функціональні вимоги клієнтської частини, що ставляться до системи

Кодовий номер	Вимога
B12	Можливість перегляду списком готових прогнозів найпопулярніших криптовалют у часових проміжках: 5хв, 10 хв, 30 хв, 1 год, 1 день, 1 тиждень, 1 місяць, 3 місяці, 6 місяців.
B13	Можливість задання власної пари криптовалют та часового проміжку для отримання персонального прогнозу
B14	Можливість задання бажаного вигляду прогнозу: таблицею чи графіком
B15	Можливість вивантаження таблиці прогнозу у Excel

Таблиця 3.3

Матриця залежності функціональних вимог до серверної частини
програмного забезпечення

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
B1	X	X	X	X	X	X	X	X	X	X	X
B2		X	X	X	X	X	X	X	X	X	X
B3			X	X	X	X	X	X	X	X	X
B4				X	X	X	X	X	X	X	X
B5					X	X					
B6						X					
B7							X	X	X	X	X
B8								X	X	X	X
B9									X	X	X
B10										X	X
B11											X

Таблиця 3.4

Матриця залежності функціональних вимог до клієнтської частини
програмного забезпечення

	B12	B13	B14	B15
B12	X		X	X
B13		X	X	X
B14			X	X
B15				X

Як видно з матриць залежності, які наведені у табл. 3.3 та 3.4 вимог жодних протиріч або накладань вимог не виявлено.

Для демонстрації структури та можливостей моделі використано універсальну мову моделювання (Unified Modeling Language, UML).

Так, можливості відображаються діаграмою варіантів використання, яку можна побачити на рис. 3.1. Нею визначаються загальні межі і контекст прогнозування курсу криптовалюти, формулюються загальні вимоги до функціональної поведінки програмного засобу, що синтезується. Як наслідок створюються передумови для подальшого його представлення логічною і фізичною моделями. Логічна структура програмного засобу прогнозування курсу криптовалют представлена за допомогою діаграм класів і діяльності – рис. 3.2 та рис. 3.3 відповідно. При цьому виокремлено послідовність дій, умов їх виконання та потік об'єктів, що необхідні для прогнозування курсу криптовалют. Архітектуру означеного засобу визначено діаграмою компонентів на рис. 3.4 та рис. 3.5. Відомості про використану платформу, інструменти, технології та обчислювальні ресурси показано діаграмою розгортання на рис. 3.6.

Таким чином, завдяки побудові об'єктно-орієнтованої моделі програмного засобу прогнозування курсу криптовалюти визначено його можливості, логічну та фізичну структури. Як наслідок, створено передумови для його програмного реалізування.

3.2. Діаграма варіантів використання

Для проектування варіантів використання необхідно визначити дійових осіб-акторів. Оскільки програмне забезпечення прогнозування курсу криптовалют ставить на меті дати можливість користувачеві задавати параметри бажаної криптовалютної пари і часового періоду прогнозу та отримувати результат у одній з двох можливих форм відображення: графіку чи таблиці – у програмному засобі єдиним учасником системи є користувач.

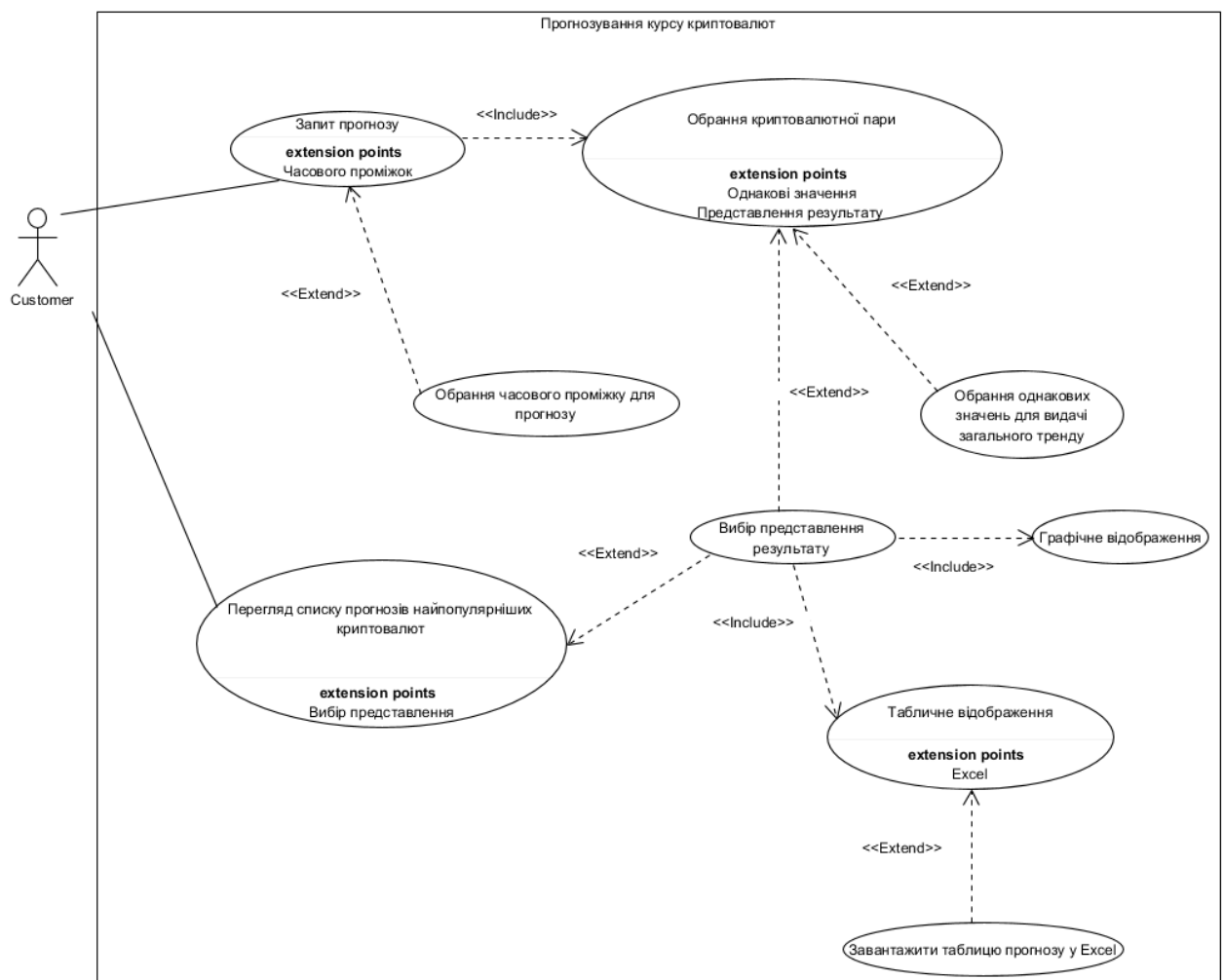


Рис. 3.1. Структурна схема варіантів використання програмного засобу прогнозування курсу криптовалют

Детальний опис кожного з варіантів використання наведено в табл. 3.5 – 3.9.

Таблиця 3.5

Варіант використання

«Перегляд списку прогнозів найпопулярніших криптовалют»

Найменування	Перегляд списку прогнозів найпопулярніших криптовалют
Первинний актор	Користувач
Інші актори	Немає
Опис	Можливість одразу побачити прогнози на стандартні часові проміжки на головній сторінці додатку без запитів
Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Відкрита головна сторінка додатку

Таблиця 3.6

Варіант використання «Задання стандартного представлення для списку прогнозів найпопулярніших криптовалют»

Найменування	Задання стандартного представлення для списку прогнозів найпопулярніших криптовалют
Первинний актор	Користувач
Інші актори	Немає
Опис	При обранні криптовалюти зі списку замість стандартного представлення графіком можна обрати табличний вигляд

Продовження табл. 3.6

Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Відкрита головна сторінка додатку зі зміненим представленням прогнозів

Таблиця 3.7

Варіант використання

«Завантаження прогнозу зі списку у вигляді Excel документа»

Найменування	Завантаження прогнозу зі списку у вигляді Excel документа
Первинний актор	Користувач
Інші актори	Немає
Опис	При обранні прогнозу з представленого списку можна натиснути на кнопку «Завантажити в Excel» для завантаження прогнозу у вигляді таблиці в документі Excel
Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Завантажено Excel документ з прогнозом

Таблиця 3.8

Варіант використання «Запит прогнозу конкретної криптовалюти»

Найменування	Запит прогнозу конкретної криптовалюти
Первинний актор	Користувач
Інші актори	Немає

Продовження табл. 3.8

Опис	Зверху на головній сторінці обираємо у випадуючих списках пару криптовалют та натискаємо на кнопку «Прогноз» для отримання прогнозу
Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Замість списку валют відображений графічний прогноз обраних криптовалют з кнопками перемикачами між стандартними інтервалами

Таблиця 3.9

Варіант використання «Запит прогнозу конкретної криптовалюти на встановлений часовий проміжок»

Найменування	Запит прогнозу конкретної криптовалюти на встановлений часовий проміжок
Первинний актор	Користувач
Інші актори	Немає
Опис	Зверху на головній сторінці обираємо у випадуючих списках пару криптовалют, після чого обираємо діапазон прогнозу зі списку діапазонів та натискаємо на кнопку «Прогноз» для отримання прогнозу
Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Замість списку валют відображений графічний прогноз обраних криптовалют на заданий часовий проміжок

3.3. Логічна модель

Логічна модель даних виражена незалежно від реалізуючих програмний засіб технологій, для надання відомостей про основні програмні сутності, у нашому випадку об'єктно-орієнтованої моделі – це класи. А також точного відображення ходу роботи програми.

Для реалізації програмного засобу потрібно було розбити дані, які ми можемо дістати з криптовалютної біржі на самодостатні одиниці, які мають роль у попиті на криптовалюту, та з яких можна скласти відповідний часовий ряд.

Також слід було обрати конкретну криптовалютну біржу, оскільки API кожної з них відрізняється. У подальшому при розширенні програми має сенс виокремити методи біржі у окремий інтерфейс, для додавання можливості роботи з іншими біржами – це підвищить працездатність програми, у разі відмови серверу однієї з бірж постачальників даних, а також буде можливо поєднувати отримані дані для заповнення пробілів інформації, які можливі через утаєння або просто відсутність деяких критеріїв на певній біржі.

Подальший механізм який дозволить це реалізувати у програмному засобі ефективно називається Dependency Injection, або Ін'єкція залежностей, про який детально буде розповідатися у четвертому розділі з описом конкретних технологій та переваг використання даного методу.

Наразі ж ми повинні розуміти, що для роботи з API біржі необхідний окремий спеціалізований на конкретній біржі клас, який представлений вкупі з іншими на рис. 3.2 зображено структурну схему класів програмного засобу, яка представлена за допомогою відповідної UML-діаграми.

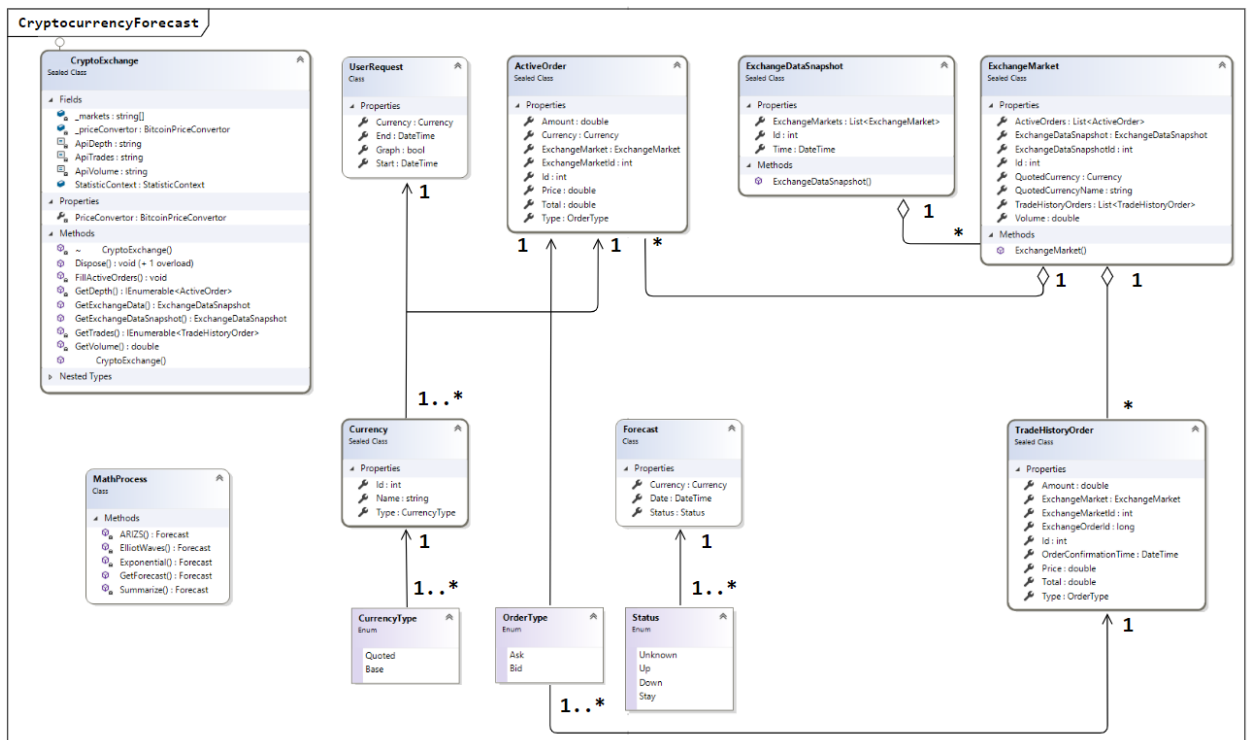


Рис. 3.2. Діаграма класів

Детальний опис кожного класу, щодо його призначення, наявних полів та методів зазначено нижче в табл. 3.10.

- *CryptoExchange* – клас який призначений для роботи з криптовалютною біржею.

Містить в собі список шаблонів HTTP-запитів до API біржі, в які підставляються дані залежно від введених користувачем значень, після чого готові запити відправляються до біржі.

Відповідь ми отримуємо у JSON форматі, який відразу ж конвертується в об'єкти програми.

Через те, що API біржі не має функції «Віддати всі дані» – процедура запиту складається з декількох підзапитів, які виконуються окремими функціями, після чого поєднуються у єдиний клас *ExchangeDataSnapshot*.

- *ExchangeDataSnapshot* – клас, який являє собою глобальну одиницю часового ряду біржових даних. Він має свій ідентифікаційний номер

та часову марку. А головне – він містить список екземплярів класу *ExchangeMarket*.

- *ExchangeMarket* – клас який містить в собі комбіновані дані зі всієї криптовалютної біржі за певний період часу, а саме:
 - список відкритих угод у вигляді екземплярів класу *ActiveOrder*;
 - котирувана валюта у вигляді екземпляру класу *Currency*;
 - список укладених угод на заданий момент часу у вигляді екземплярів класу *TradeHistoryOrder*;
 - поточний об'єм торгів зберігається у змінній *Volume*.
- *ActiveOrder* – клас який відображає відкриту угоду на біржі, з такими полями як:
 - ціна;
 - кількість;
 - тип угоди;
 - сумарна сума угоди;
 - базова валюта у вигляді екземпляру класу *Currency*.
- *Currency* – клас, який просто інкапсулює в собі ім'я та ідентифікаційний номер криптовалюти, вкупі з її поточним типом:
 - базова валюта – валюта в конкретній валютній парі, ціна однієї одиниці якої завжди міряється в одиницях іншої (котирується) валюти.
 - котирувана валюта – валюта, в одиницях якої виражається ціна однієї одиниці базової валюти.
- *TradeHistoryOrder* – клас який відображає закриту угоду на біржі, з полями:
 - ціна;
 - кількість;
 - тип угоди;

- сумарна сума угоди;
- час закриття угоди.
- *UserRequest* – клас який представляє запит користувача та містить поля:
 - базова валюта у вигляді екземпляру класу *Currency*;
 - котирувана валюта у вигляді екземпляру класу *Currency*;
 - початок заданого часового діапазону у типі *DateTime*;
 - кінець заданого часового діапазону у типі *DateTime*;
 - булеве значення яке визначає у якому вигляді повинен бути виданий прогноз.
- *MathProcess* – статичний клас який містить в собі статичні методи для обробки даних математичним алгоритмом чи передачі для обробки в модуль нейронної мережі з відповідними методами, всі вони повертають клас *Forecast*;
- *Forecast* – клас який містить в собі дані прогнозу по тренду чи курсу з часовим рядом.

Як можна побачити на діаграмі присутні ще три типи, які є введеними переліченнями, які є у кожній сучасній ООП мові. Вони допомагають об'єднати набір числових значень за логічним відношенням, та надати ім'я кожному з них. Таким чином можна використовувати їх у класах як розширені параметри ознак з зрозумілими іменами.

До таких типів на діаграмі відносяться:

- *CurrencyType* – перелічення для визначення типу валюти з двома полями:
 - *Quoted* – яке позначає котирувану валюту;
 - *Base* – яке позначає базову валюту.
- *OrderType* – перелічення для визначення типу угоди з двома полями:
 - *Ask* – заява для угоди на продаж;
 - *Bid* – заява для угоди на покупку.

– *Status* – перелічення для визначення напрямку тренду у прогнозі з полями:

- Unknown – початкове непроініціалізоване значення на випадок помилкових ситуацій;
- Up – позначає напрямок тренду вгору;
- Down – позначає напрямок тренду вниз;
- Stay – позначає, що тренд залишиться незмінним.

Після опису класів програмного засобу слід створити діаграму діяльності, яка демонструє конкретну логічну послідовність робочого циклу програми. Створену діаграму можна побачити на рис. 3.3.

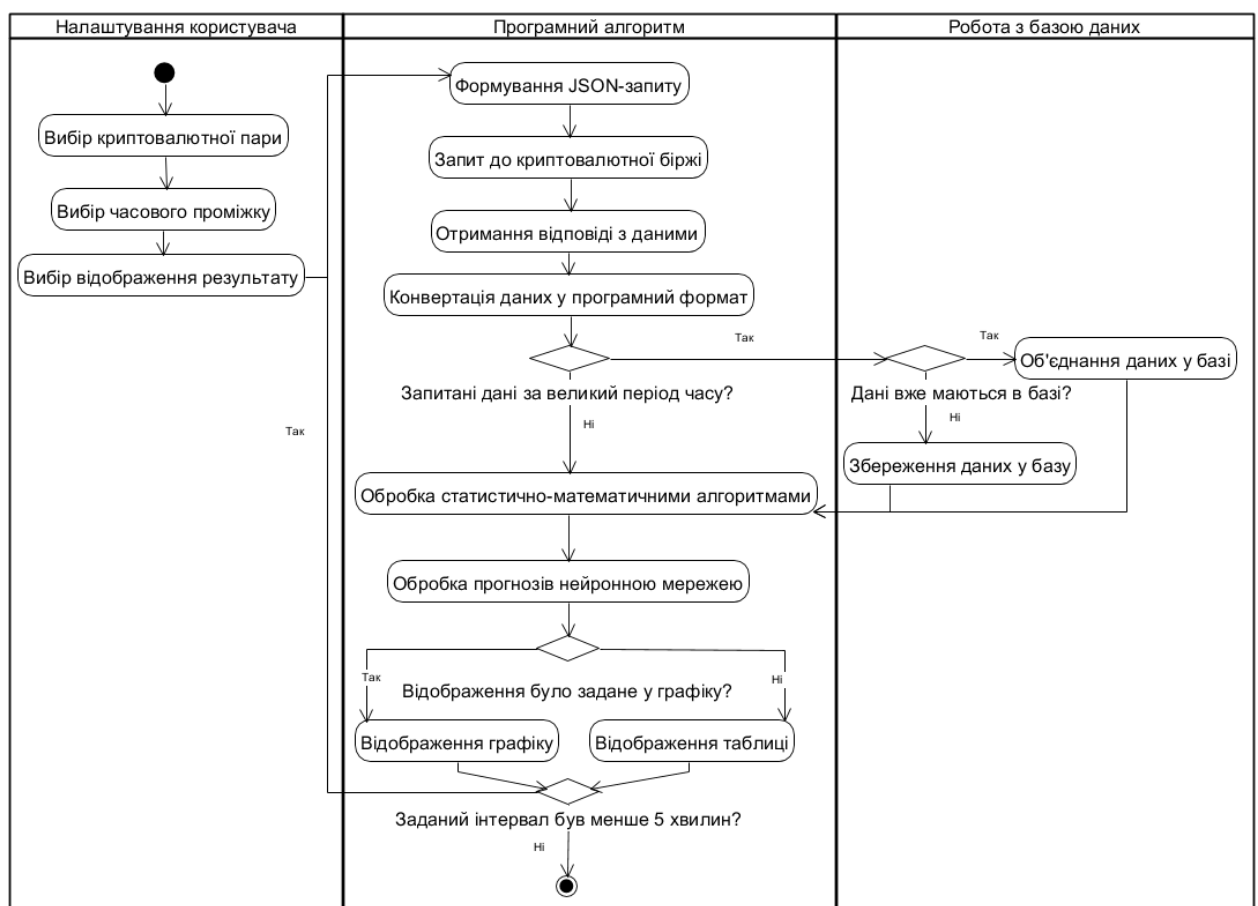


Рис. 3.3. Діаграма діяльності

У діаграмі слід зазначити можливість програмного циклу, при умові, що користувач запитав прогноз на строк менший за 5 хвилин, в даному

випадку замість відображення статичного результату прогнозу програма може повторювати запит до криптовалютної біржі для оновлення результатів в режимі реального часу.

Інакше кажучи – користувач матиме змогу спостерігати за рухом графіку курсу на екрані, як на звичайній біржі, але з тією відмінністю, що це буде прогноз ціни.

3.4. Фізична модель

Для визначення фізичної структури програмного засобу прогнозування курсу криптовалют було побудовано діаграми компонентів, які зображені на рис. 3.4 – 3.5.

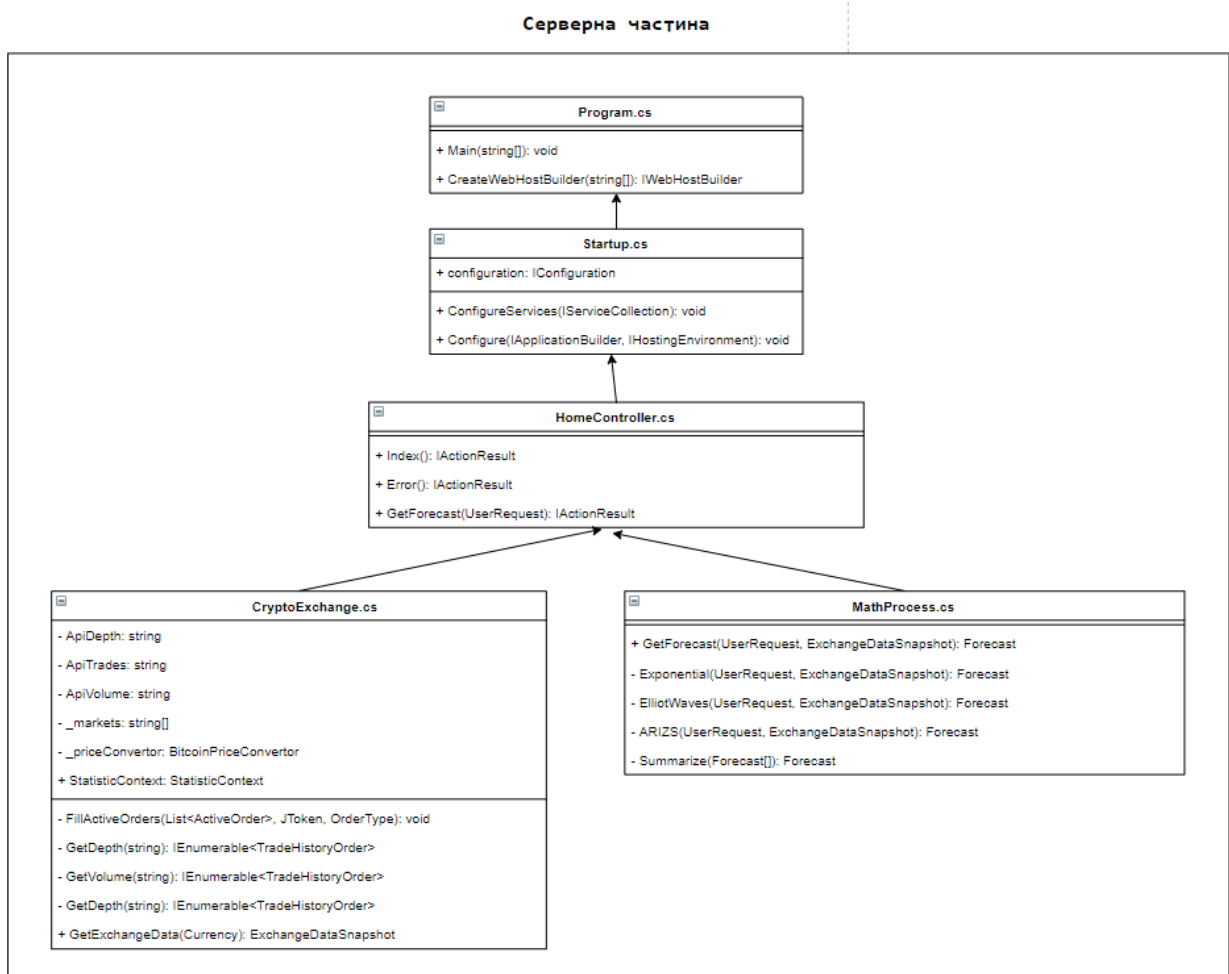


Рис. 3.4. Діаграма компонентів серверної частини

На рис. 3.4. продемонстровано структура файлів програмного засобу прогнозування криптовалют, а конкретно серверної частини, які дають можливість опрацюванню всіх даних та роботи програми як було показано на рис. 3.3.

- Program.cs – клас програми який є точкою входу для початку виконання програми;
- Startup.cs – головний клас програми, який забезпечує роботу системи завдяки початковій конфігурації усіх наявних сервісів, таких як MVC, тощо.
- HomeController.cs – клас програми який відповідає за взаємодію з користувачем через головну сторінку додатку. В ньому:
 - користувачу надається початковий список прогнозів;
 - користувачу надається форма для запиту конкретного прогнозу з додатковими полями фільтрами для вибору криптовалютної пари та часового діапазону прогнозу;
 - отриманий від користувача запит надсилається до модуля обробки;
 - отриманий прогноз передається саме в цей контролер для його подальшого відображення як відповіді користувачу.
- CryptoExchange.cs – клас, який необхідний для роботи з API криптовалютної біржі, його було детально описано в логічній моделі та на діаграмі класів;
- MathProcess.cs – статичний клас, який відповідає за всю обробку даних математично-статистичними методами та ШНМ для видачі прогнозів.

Потрібно відмітити, що вищезазначений список не містить файлів з класами-моделями, які не виконують жодних дій, а являються лише об'єктним представленням даних переданих нам у вигляді JSON з

криптовалютної біржі та використовуються для передачі та зберігання числових даних.

Далі для роботи безпосередньо з клієнтом використовуються файли з розміткою, такі як:

- .html чи .chtml – застосовуються для розмітки у веб-додатках за допомогою мови HTML та вставок в неї мови JavaScript чи C#;
- .haml – застосовуються для розмітки у десктопних додатках розроблених на платформі .NET.

В клієнтській частині програмного засобу прогнозування криптовалют передбачені файли продемонстровані на рис. 3.5.

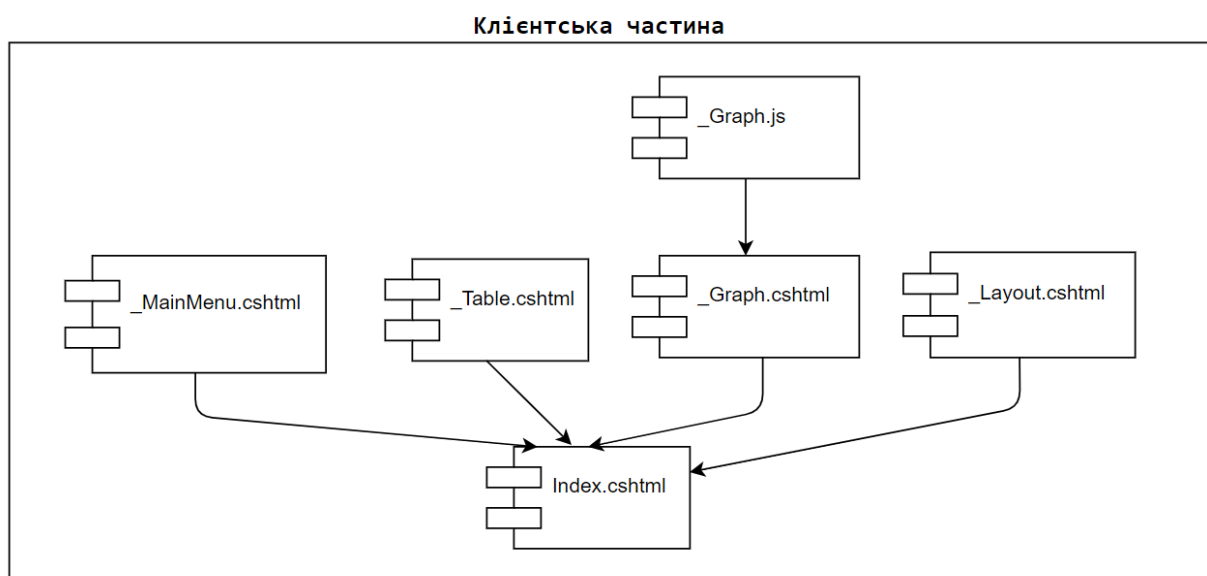


Рис. 3.5. Діаграма компонентів клієнтської частини

Клієнтська частина складається з наступних файлів:

- Index.chtml – являє собою головну та єдину сторінку додатку, оскільки додаток планується як односторінковий застосунок (SPA).

Це означає, що саме ця сторінка з усіма наявними скриптами завантажується користувачу, а подальші дії виконуються за допомогою асинхронних вставлень часткових представлень до сторінки.

- `_MainMenu.cshtml` – часткове представлення яке вбудовується у головну сторінку `Index.cshtml` та надає користувачу меню, де він може задавати параметри запиту прогнозу та форму його відображення;
- `_Layout.cshtml` – містить загальний шаблон сторінки та установки щодо розміщення елементів;
- `_Table.cshtml` – часткове представлення яке вбудовується у головну сторінку `Index.cshtml` та відображає прогноз у вигляді таблиці з датою, часом та курсом криптовалюти;
- `_Graph.cshtml` – часткове представлення яке вбудовується у головну сторінку `Index.cshtml` та містить скрипт `_Graph.js` для вбудови графіку в нього;
- `_Graph.js` – скрипт на мові JavaScript, який використовує інструменти відкритих бібліотек для візуалізації даних прогнозу у вигляді графіку.

Для повноти фізичної моделі необхідно побудувати діаграму розгортки, яка визначає конкретні інструменти, якими будуть реалізовані поставлені вимоги та продемонстровані моделі.

Програмний засіб прогнозування криптовалюти передбачає можливість експлуатації декількома користувачами, кожен з яких буде взаємодіяти з власним наданим інтерфейсом. З цією ціллю діаграма на рис. 3.6 розроблена з точки зору веб-додатку, а користувач буде взаємодіяти з ним через веб-браузер. Вибір даного способу реалізації детально описаний в наступному розділі. На роль криптовалютної біржі було обрано компанію YoBit.Net, яка має широкий діапазон даних, що надається через API у форматі JSON, та має високий рівень довіри серед інтернет спільноти.

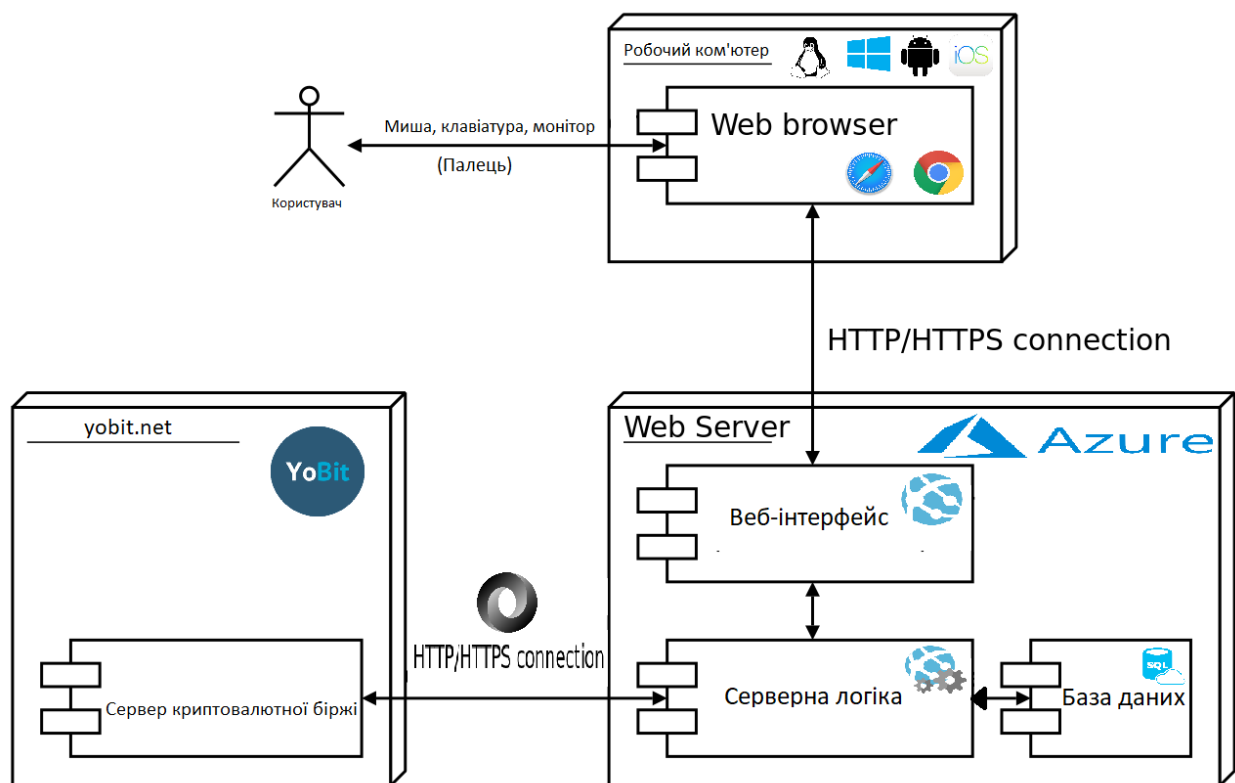


Рис. 3.6. Діаграма розгортки

Діаграма складається з наступних компонентів:

- робочий комп'ютер на будь-якій операційній системі та веб-браузер через який користувач взаємодіє з сервером програмного додатку;
- веб-сервер, на якому розгорнуті:
 - веб-додаток – на основі ASP.NET MVC Core;
 - база даних – на основі Azure SQL Cloud Database.
- криптовалютна біржа представлена компанією YoBit.Net за інтернет посиланням <https://yobit.net>

3.5. Висновки

Було розглянуто та побудовано всі необхідні діаграми для чіткої концепції програмного засобу для прогнозування криптовалют у концепції ООП.

Побудовані діаграми:

- варіантів використання;
- класів;
- діяльності;
- компонентів;
- розгортки.

Завдяки цьому ми було продумано кожний етап та механізм роботи програмного засобу і стало можливим перейти до його реалізації.

4. ПРОГРАМНИЙ ЗАСІБ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ

Архітектуру означеного засобу визначено діаграмою компонентів. Відомості про платформу, обчислювальні ресурси показано діаграмою розгортання. При цьому як найпростіший варіант для розповсюдження, підтримки та кросплатформності програмного засобу застосовано веб-форму його представлення. Виходячи з необхідної обчислювальної потужності обрано платформу .NET. Це дозволило створити серверну частину, інтерфейс, базу даних та виконати розгортання програмного засобу на платформі Azure.

Таким чином, завдяки побудованій об'єктно-орієнтованій моделі розроблено програмний засіб прогнозування курсу криптовалюти. Для цього використано платформи .NET та Azure і, як наслідок, отримано веб-застосунок з хостингом на хмарному сервері та базовим веб-інтерфейсом у вигляді таблиць, графіків та фільтрів криптовалют.

4.1. Аналіз вимог до інструментів для реалізації програмного засобу

Першочергова ціль – це дослідити ефективність створеного методу в рамках реальних торгів на криптовалютній біржі. Отже насамперед необхідно обрати такі інструменти та набір технологій, з якими можлива швидка реалізація та розгортка працюючого додатку.

А це в свою чергу можливе при широкому набору готових бібліотек та фреймворків, які пришвидшують розробку завдяки готовим компонентам.

При цьому ми вже маємо детальні розплановані схеми та діаграми ООП моделі нашого програмного засобу, отже слід обрати об'єктно-орієнтовану мову програмування та середовище, яке її підтримує.

Наступним пріоритетом йде швидкодія розробленого додатку, яка дозволить в повній мірі реалізувати математично-статистичні оброблювані

алгоритми, а також дозволить нейронній мережі працювати в режимі реального часу без суттєвих затримок.

Далі, нам необхідний зручний та мінімальний інтерфейс, який буде відповідати концепції мінімально працюючого продукту, але при цьому мати всі необхідні функції, які були оговорені у поставлених задачах та діаграмі використань, а це наявність виводу графіків і таблиць, а також списки та інтервальна шкала часу для параметризації запиту.

Рішення також має бути якомога атомарнішим, для зручності як розробки, так і подальшої підтримки для покращення програмного засобу, оскільки при роботі з нейронними мережами та алгоритмами прогнозів постійно будуть аналізуватися наявна результативність, для оптимізації компонентів і роботи.

Останній момент – програмний засіб має бути якомога легким для розгортання у будь-якому середовищі, операційній системі, тощо.

Виходячи з даних положень, ми маємо обрати між реалізацією програмного засобу як:

- мобільного додатку;
- десктопного додатку;
- веб-додатку.

У разі реалізації мобільного додатку ми маємо два шляхи, це розробка під мобільну операційну систему Android, чи розробку під iOS. В будь-якому випадку наш додаток буде працювати тільки на одній з них, і це звичайно мінус.

Ще один недолік – можлива недостатність апаратної потужності на багатьох сучасних смартфонах для паралельної та швидкої обробки такого об'єму даних.

У разі реалізації програмного засобу як десктопного додатку у нас є перевага у більшій обчислювальній здатності, але залишається недолік обирання операційної системи, в свою чергу їх тут вже три:

- Windows.
- OSx.
- Linux.

Хоча багато засобів для створення програмного забезпечення є кросплатформеними – вони все ще платять ціну стабільності та обмежень за неї.

Ще один загальний недолік цих двох підходів полягає у тому, що програма прив'язана до конкретного пристрою. У разі бажання його заміни – ми повинні будемо повторити процес дистрибуції та розгортки, що є втратою часу.

У випадку реалізації системи як веб-додатку – ми маємо доступ до роботи з ним з будь-якої платформи, та будь-якого пристрою з підключенням до інтернету.

Обчислення проходять на серверному комп'ютері, який, зазвичай має гарні показники потужності та потенціал до її нарощення.

Також ми звільняємося від необхідності дистрибуції та підтримки роботоспроможності додатку – все це завжди тільки обслуговування єдиного сервера.

Ще одна перевага – це можливість хмарного хостингу, при якому ми можемо взагалі не хвилюватися про апаратну частину, а при бажанні і про налаштування додатку у операційній системі – хмарні сервіси все це надають і роблять автоматично. А по мірі зростання необхідних ресурсів для роботи програми – завжди є можливість розширення просто за придбання нового користувацького плану.

Як висновок, ми маємо обрати інструменти які:

- Мають широку базу бібліотек та фреймворків;
- Підтримують концепцію ООП;
- Надають можливість розробки веб-додатку;
- Надають можливість розгортки додатку у хмарних сервісах.

4.2. Обрання платформи та стеку технологій для реалізації програмного засобу

З обраних необхідних можливостей інструментів розробки ми маємо вибір між реалізацією додатку за допомогою декількох наборів технологій.

Та єдиний вірний спосіб реалізації подібних програмних засобів, які потребують як значної швидкодії на серверній частині, так зручної розробки веб-інтерфейсу до неї з послідуною розгорткою у хмарному сервісі – це використання платформи .NET.

Вона має передову ООП мову програмування C#, яку створювали ведучі спеціалісти галузі інформаційних технологій, яку з року в рік розвивала компанія, яка лідує у сфері ІТ послуг – Microsoft.

Поточна версія C# втілила у собі багато корисних можливостей запозичених з функціональних мов програмування, мова має елегантний та легкий синтаксис, багато розширень та синтаксичного цукру.

Сама платформа .NET має пакетний бібліотечний менеджер NuGet, який дозволяє легко підключати та користуватися перевагами тисяч бібліотек для розробки.

А найголовніше – вона поєднує в собі всі етапи розробки:

- створення проекту;
- реалізація його інтерфейсу;
- розгортка у хмарному сервісі.

Останній пункт можливий завдяки присутності у .NET платформи Microsoft Azure, яка надає як хостинг для веб-додатків, так і хмарні SQL та NoSQL сервери і бази даних для повноцінного функціонування програми.

Слід зазначити, що в області математичних обчислень часто прийнято використовувати мову програмування Python, яка має найбільшу кількість

бібліотек створених саме для цілей наукової праці, та роботи з великими даними.

Але попри це сама мова є застарілою та незручною для тривалої розробки.

Саме тут приходить на поміч Dynamic Language Runtime (DLR) від .NET, яка надає змогу легкої та ефективної співпраці строго типізованої мови C# компільованої у середовищі виконання байт коду Common Language Runtime (CLR) з такими динамічними мовами як Python.

Іншими словами, ми можемо використовувати всі існуючі переваги та потужність математичних бібліотек Python'у, без необхідності безпосередньої розробки на ньому (яка була б важка та нестерпна).

А для створення веб-інтерфейсу ми застосуємо кращий веб-фреймворк на ринку – ASP.NET MVC Core.

Він має Open Source структуру, тобто доступ до його коду має все ІТ співтовариство, яке постійно виправляє помилки, пропонує покращення, та оптимізує наявні механізми.

Фреймворк є кросплатформним, але навіть це неважливо, оскільки для додатків реалізованих на ньому – хмарна платформа Azure надає послугу «Програма як послуга», або SaaS.

Тобто ми можемо розгорнути наш веб-додаток одразу на платформі, не вникаючи у деталі з операційною системою та іншим.

Також він підтримує зручну інтеграцію з усіма новими технологіями у веб розробці, такі як HTML5, CSS3, Angular, React.js, Vue.js, AJAX, тощо.

І на довершення швидкодії надає середа розробки Visual Studio, яка наразі має нову 2019 версію, і кожен етап створення додатку ми можемо контролювати безпосередньо з неї, не змінюючи вікна.

4.3. Реалізація програмного засобу

Було створено веб-додаток за допомогою інструментів платформи .NET у середовищі розробки Visual Studio 2019 на ООП мові програмування C#.

Додаток реалізовано на основі фреймворку ASP.NET MVC, який передбачає наяву:

- моделей – якими виступають описані у попередньому розділі класи у діаграмі класів сутностей.
- представлень – якими є описані в діаграмі компонентів на стороні клієнта файли з розміткою html з вставками C# коду;
- контролера – описаного в діаграмі компонентів на стороні серверу єдиного контролера, який контролює взаємодію моделей, обробки даних та представлень.

Першим етапом було створення класу для роботи з API біржі, ним став клас YoBitCryptoExchange, запити якого створювалися на основі трьох URI рядків до яких додавалося значення криптовалютної пари:

- ApiDepth – URI рядок зі значенням "https://yobit.net/api/3/depth/" для запиту поточних відкритих заявок, представлених в програмі класом ActiveOrder;
- ApiTrades – URI рядок зі значенням "https://yobit.net/api/3/trades/" для запиту закритих угод за минулий період часу, представлених в програмі класом TradeHistoryOrder;
- ApiVolume – URI рядок зі значенням "https://yobit.net/api/3/ticker/" для запиту загального об'єму продаж на ринку.

Найважливіші компоненти програмного засобу – це дві ШНМ навчені за допомогою методу зворотнього поширення помилки.

Перша ШНМ навчалася на вибірках по даним часового ряду отриманих з API біржі.

Друга навчалася по вибіркам з отриманих результатів після обробки методом Марковських випадкових процесів і ланцюгів та першою ШНМ.

Архітектура обох ШНМ відповідає багатошаровому перцептрону, її можна побачити на рис. 4.1.

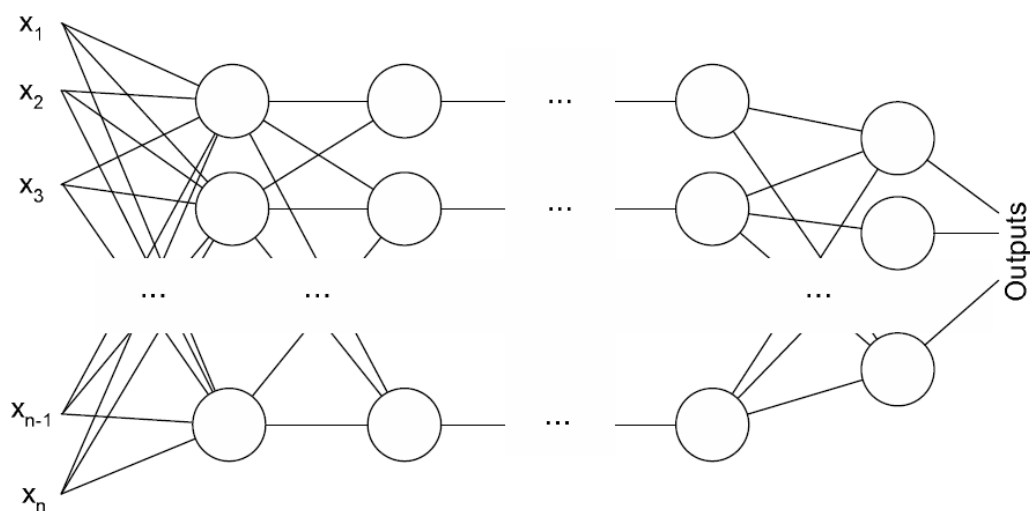


Рис. 4.1. Архітектура багатошарового перцептрону

Для вирішення задачі прогнозування часових рядів навчання ШНМ не має математичного рішення та проводилося імпіричним шляхом.

Проведено чисельний експеримент, який показав, що оптимальне значення параметрів швидкості навчання і моменту для розглядуемого часового ряду $n_{\text{опт}}, \alpha_{\text{опт}} \in (0.1; 0.5)$. Коли параметр моменту $\alpha = 0$, оптимальне значення параметра швидкості навчання $n_{\text{опт}} \rightarrow 1$.

Також кінцева середньоквадратична помилка мало відрізняється для різних кривих, що означає доволі гладку поверхню помилок в розглянутій нами задачі прогнозування.

Згідно проведенням тестам було вирішено, що оптимальна кількість прихованих шарів дорівнює двом.

Вхідний шар складається з 5 нейронів. На першому прихованому шарі знаходиться 10 нейронів, а на другому – 15. Вихідний шар має 1 нейрон.

Значення основних параметрів ШНМ моделі наведені в табл. 4.1.

Таблиця 4.1

Значення основних параметрів ШНМ моделі

Назва параметра	Значення параметра
Діапазон початкових значень вагових коефіцієнтів	$ w_{ji}^{(0)} < 1$
Коефіцієнт швидкості навчання η	0.5
Момент α	0.1
Точність навчання ε	0.005

Дизайн веб-сторінок забезпечено вбудованими стилями Bootstrap, які адаптивно налаштовані для оформлення простих елементів на кшталт меню, кнопок, таблиць та випадаючих списків.

Графік курсу отриманого прогнозу відображається завдяки відкритій бібліотеці «Awesome Chart.js».

Кешовані біржові дані та прогнози зберігаються у хмарній базі даних, яка налаштована паралельно з веб-додатком, та має параметр до автоматичного розширення потужностей, при необхідності.

Тобто і база даних, і програма мають можливість горизонтального розширення для паралельної роботи з великою кількістю користувачів. Обмеження лише в бюджеті.

4.4. Тестування розробленого програмного засобу

Після завершення навчання ШНМ на прикладах з навчальної множини якість прогнозів перевіряється на прикладах тестової множини. Для цього вираховується помилка прогнозування нейронної мережі за формулою (4.1).

$$E = \frac{1}{N} \sum_{t=1}^N \frac{(z_t - \hat{z}_t)}{z_t} \cdot 100\%, \quad (4.1)$$

де $N = 25$ – розмір навчальної множини. \hat{z}_t – прогнозоване значення для t -го тестового значення z_t .

Для ШНМ, навченої за алгоритмом зворотнього поширення помилки середні помилка прогнозування на тестовій множині $E = 0.4487\%$.

Точність прогнозу напрувлення тренда склала 52%.

Результати прогнозів ШНМ для первинної обробки даних часового ряду з біржі приведені на рис. 4.2.

Для ШНМ навченої для комбінування прогнозів на основі вхідних даних з інших прогнозів середня помилка прогнозування на тестовій множині склала $E = 0.4106\%$, точність прогнозу напрямку тренду склала 76%. Результати прогнозування для другої ШНМ приведені на рис. 4.3.

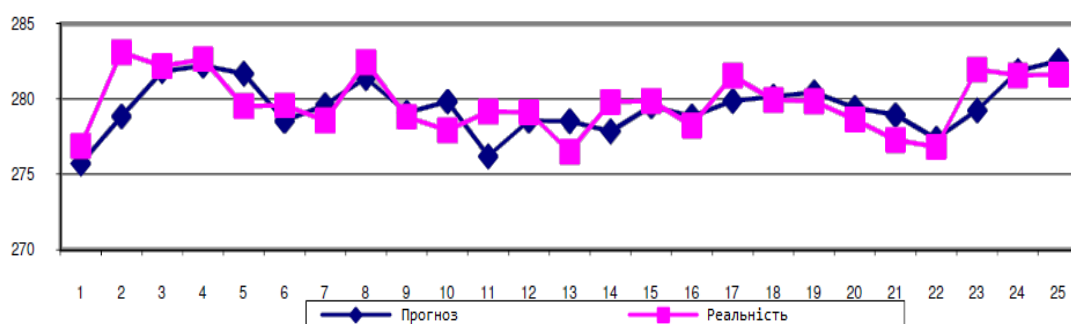


Рис. 4.2. Порівняння прогнозу з реальністю ШНМ для первинної обробки даних

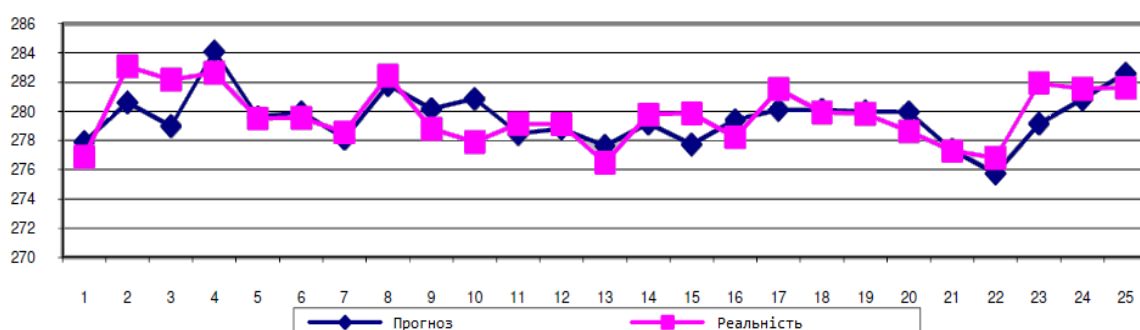


Рис. 4.3. Порівняння прогнозу з реальністю ШНМ для комбінування прогнозів

Після проведення локальних тестів додаток був запущений для роботи в режимі реального часу, що продемонстровано на рис. 4.4.

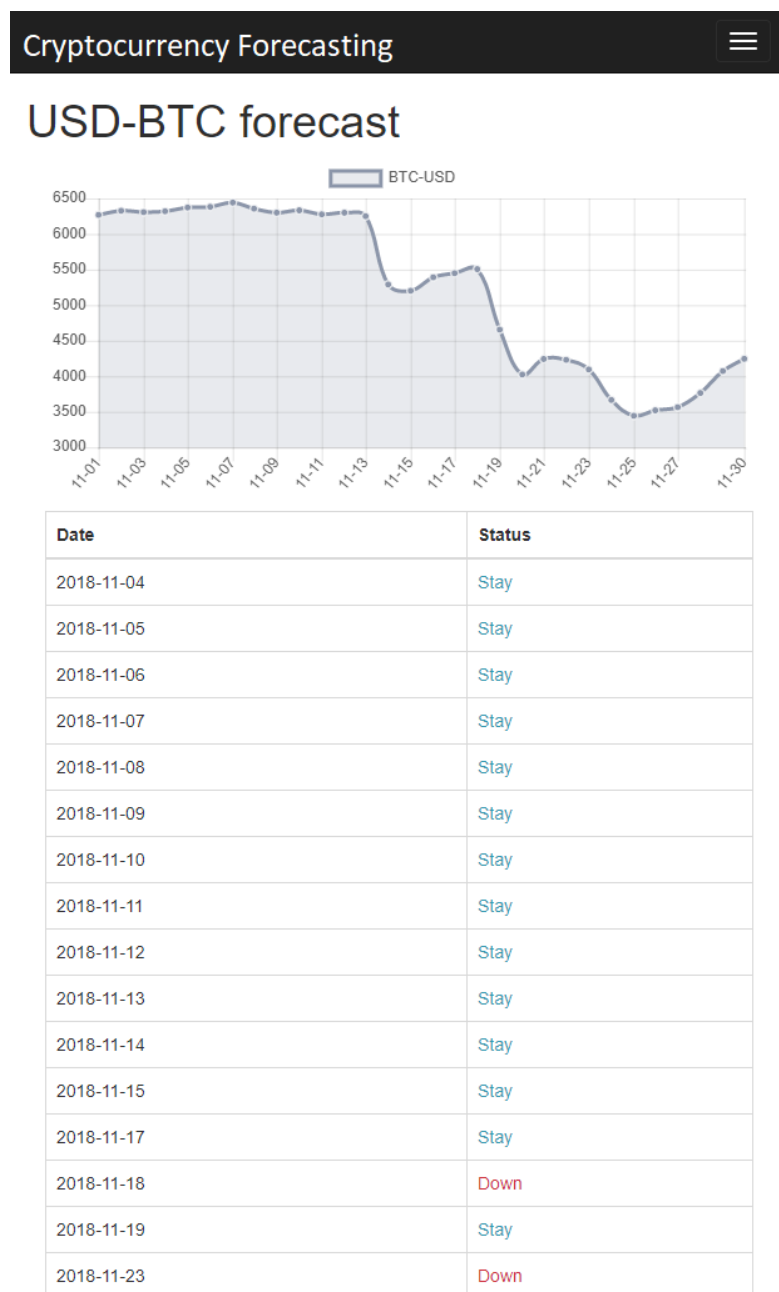


Рис. 4.4. Скріншот роботи веб-додатку прогнозування курсу криптовалют

4.5. Висновки

У даному розділі було повністю описано вибір програмних технологій виходячи з поставлених умов у попередніх розділах, та відштовхуючись від зробленого аналізу наявних рішень, як результат було обрано реалізувати програмний засіб прогнозування криптовалют за допомогою технологій платформи .NET.

Створено веб-застосунок на основі ASP.NET MVC, написаний на мові С#. Додаток було розгорнуто на хмарній платформі Microsoft Azure, разом з супутньою базою даних.

Для створення тестових та навчальних наборів, і подальшої роботи додатку як джерело даних було обрано криптовалютну біржу YoBit.Net.

Тестування навчених ШНМ показало:

- точність прогнозу вище середнього для першої мережі, яка працювала з часовим рядом даних біржі;
- висока точність прогнозу для другої мережі, яка працювала з часовим рядом прогнозів зроблених першою мережею та методом на основі Марковських випадкових процесів і ланцюгів.

Програма була успішно запущена в режимі реального часу.

ВИСНОВКИ

Було виконано аналіз актуальності задачі прогнозування курсу криптовалют, яка підтвердила високу популярність з боку користувачів криптовалют, та трейдерів.

Розглянуто існуючі методи для вирішення задач прогнозування курсу валют та криптовалют, які сьогодні використовують трейдери та спеціалізовані програми для аналізу ринку; проведено їх детальний аналіз щодо зручності, часозатрат, типу даних та ефективності прогнозу.

У результаті чого зроблено висновок про їх недостатню ефективність.

Запропонований оптимізований метод прогнозування і розглянуто структурну модель програмного засобу прогнозування криптовалюти на основі запропонованого методу у нотації IDEF0, та її декомпозиціях у нотаціях IDEF0, IDEF3 і DFD.

Запропоновано об'єктно-орієнтовану модель програмного засобу прогнозування курсу криптовалюти з описом діаграми варіантів використання, логічної та фізичної моделі.

Проаналізовано вимоги до існуючих засобів розробки програмного забезпечення для реалізації методу. Обрано та обґрунтовано вибір стеку технологій, та описано як за його допомогою було створено програмний засіб прогнозування криптовалют. Отриманий засіб було протестовано на працездатність та ефективність.

Тести показали підвищення точності прогнозів завдяки доданню комбінуючої ШНМ, як описано в запропонованому методі.

Що надає причини вважати метод та реалізуючий його програмний засіб прогнозування курсу криптовалют успішним та ефективним.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Фёдорова Е.А. Прогнозирование курса валюты с помощью нейронных сетей / Е.А. Фёдорова, М.А. Линкова // Финансовая аналитика: проблемы и решения. – 2013. – 11, 149. – С. 27-31.
2. Arvind Narayanan. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction / Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller & Steven Goldfeder // Princeton University Press. 2016 – 304 с.
3. Ivan Nunes da Silva. Artificial Neural Networks: A Practical Course / Ivan Nunes da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, Silas Franco dos Reis Alves // Springer. 2017. – 307 с.
4. Tariq Rashid. Make Your Own Neural Network / Tariq Rashid // CreateSpace Independent Publishing Platform. 2016. – 222 с.
5. Adem Efe Gencer. Decentralization in Bitcoin and Ethereum Networks / Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, Emin Gün Sirer // Financial Cryptography and Data Security (FC). 2018. – 18 с.
6. Анализ временных рядов [Электронный ресурс]. – Режим доступа: <http://statsoft.ru/home/textbook/modules/sttimser.html#arima>
Дата доступа : Травень 2019. – Назва з екрана.
7. Основи технічного аналізу [Електронний ресурс]. – Режим доступу : https://pidruchniki.com/72559/finansi/osnovi_tehnichnogo_analizu.
Дата доступу : Травень 2019. – Назва з екрана.
8. Технічний аналіз фондового ринку [Електронний ресурс]. – Режим доступу : https://stud.com.ua/50044/finansi/tehnichniy_analiz_fondovogo_rinku.
Дата доступу : Травень 2019. – Назва з екрана.
9. YOBIT.API [Електронний ресурс]. – Режим доступу: <https://github.com/nettrash/YOBIT.API>

Дата доступу: Травень 2019. – Назва з екрана.

10. *Панілов М.А.* Економічне моделювання динаміки валютного курсу. / Автореф. дисертація канд. екон. наук. Москва, 2009. – 32 с
11. *П Колбі Р.В., Мейерс Т.А.* Енциклопедія технічних індикаторів: Пер. з англ. М. / Видавничий будинок «Альпіна», 1998. – 581 с.
12. *Якімкін В.Н.* Як почати заробляти на валютному ринку Forex. М. / Смартбук, 2008. – 352 с.
13. *Ричков В.В.* Удосконалення інструментальних методів аналізу та прогнозування міжнародного валютного ринку. / Автореф. дисертація канд. екон. наук. Пермь, 2001. – 22 с.
14. *Зінін А.Н.* Моделі прибутковості і прогнозування ризику портфеля інвестора на міжнародному валютному ринку (на прикладі ринку Forex). / Автореф. дисертація канд. екон. наук. Ростов-на-Дону, 2003. – 24 с.
15. *Літінський Д.С.* Статистичне прогнозування для побудови ефективних торгових стратегій на валютному ринку. / Автореф. дисертація канд. екон. наук. Москва, 2003. – 23 с.
16. *Болотова Л.Р.* Математичні методи статистики і нелінійної динаміки для оцінки валютних ризиків на базі передпрогнозного аналізу. / Автореф. дисертація канд. екон. наук. Ставрополь, 2005. – 23 с.
17. *Смірнов С.В.* Статистичні моделі аналізу факторів, що впливають на динаміку валютних курсів. / Автореф. дисертація канд. екон. наук. Санкт-Петербург, 2005. – 16 с.
18. *Гуляєва О.А.* Управління валютними ризиками на основі передпрогнозного аналізу валютних курсів фрактальними методами. Автореф. дисертація канд. екон. наук. Москва, 2008. – 27 с.
19. *Сельцовський В.Л.* Економіко-статистичні методи аналізу зовнішньої торгівлі - М. Фінанси і статистика, 2004. - 512 с.
20. *Dr Garrick Hileman, Michel Rauch.* Global cryptocurrency benchmarking study / Cambridge Centre for Alternative Finance, 2017. – 114 с.

21. *Michael Parkin*. The Addison-Wesley Series in Economics, Addison-Wesley, 2004. – 840p.
22. *Victor Niederhoffer*. Practical Speculation. Wiley, 2007. – 400 c.
23. *Mak D.K.* Science of Financial Market Trading. World Scientific Publishing Co. Pte. Ltd., Singapore, 2003. – 260p.
24. *Mak D.K.* Mathematical techniques in financial market trading. World Scientific Pub Co Inc, 2006. – 320p.

Додатки

Додаток 1
Копія презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП’ЮТЕРНИХ СИСТЕМ

Метод і програмний засіб прогнозування курсу криптовалюти

Виконав : Яхін Сергій Леонідович

Керівник: Цуркан Василь Васильович, Доцент, к.т.н.

Київ – 2019



ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об'єкт дослідження: процес прогнозування курсу криптовалют.

Предмет дослідження: метод і програмний засіб прогнозування курсу криптовалюти.



МЕТА РОБОТИ ТА МЕТОД ДОСЛІДЖЕННЯ

Мета роботи: створити метод прогнозу курсу криптовалюти, реалізувати його у вигляді програмного засобу та дослідити ефективність.

Метод дослідження: розгляд та аналіз методів прогнозування курсу криптовалют: Марковські випадкові процеси та ланцюги Маркова, метод експоненціального згладжування з поєднанням результатів за допомогою багатосарового перцептрона Румельхарта.



АКТУАЛЬНІСТЬ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАСОБУ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

Існує понад 2500 криптовалют. Це ринок, що швидко розвивається, і який складно регулюється в багатьох країнах світу. Іноваційність, великі перспективи та зацікавленість світової спільноти робить криптовалюту цінним активом, за який активно ведуться торги на біржах. З огляду на це, зростає потреба в їх прогнозуванні.



Існуючі методи ТА мають обмежені результати точності прогнозу, та можуть бути ефективно застосовані лише експертами у поєднанні з методами фундаментального аналізу.

Даний метод має на меті заміну фінального підсумку прогнозу людиною на прогноз від ШНМ з архітектурою багат шарового перцептрона Румельхарта, яка діятиме на основі даних наданих методами ТА.



СУТНІСТЬ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

Прогнозування курсу є процесом визначення майбутнього курсу за набором пов'язаних з ним значень: попереднього курсу, активності угод, їх типу, часу та вартості.

Для прогнозування використовуються формалізовані, евристичні та комплексні методи прогнозування, з використанням різних підходів збору інформації: науковий аналіз, визначення і аналіз причинно-наслідкових та інших зв'язків між ними.

У результаті є значення чи їх діапазон на певний період часу.



Задачі

1. Провести аналіз існуючих методів та практик прогнозування курсу криптовалюти;
2. Розробити метод оптимізуючий процес прогнозування курсу криптовалюти;
3. Розробити концептуальну модель програмного засобу прогнозування курсу криптовалюти;
4. Розробити об'єктно-орієнтовану модель програмного засобу прогнозування курсу криптовалюти;
5. Обрати форму репрезентації програмного засобу прогнозування курсу криптовалюти;
6. Обрати стек технологій для реалізації програмного засобу прогнозування курсу криптовалюти;
7. Розробити програмний засіб прогнозування курсу криптовалюти;
8. Дослідити реалізований програмний засіб прогнозування курсу криптовалюти на ефективність.



АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ПРОГНОЗУВАННЯ КУРСУ ВАЛЮТ

	Інтерфейс	Точність	Зручність	Функціональність
Trader	DOS	Похибка 10-50%. Задовільно.	Відсутність довідки, погане меню, гарячі клавіши	Ковзне середнє : лінійне, експоненціальне, з задаваними вагами; MACD – гістограми; Індикатори: RSI, OBV, Williams R%, CandleSticks, Point & Figure.
EWAP	Windows	Похибка 8-45% Добре.	Довідка з теоретичним описом ТХЕ, розділ з усіма етапами аналізу даних по ТХЕ.	Обчислення методом Теорії Хвиль Елліота (ТХЕ).
AINET	Windows	Похибка 15-55%. Задовільно.	Відсутність графіків, тільки таблиця з числами. Принцип роботи по чорному ящику. Дуже просте меню.	Нейронні мережі виду багатoshарового перцептрону.

АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ КУРСУ ВАЛЮТ



МОДЕЛІ АВТО-РЕГРЕСІЇ ІНТЕГРУВАННЯ ЗМІННОГО СЕРЕДНЬОГО (АРІЗС)

Переваги	Недоліки
Математико-статистичне обґрунтування	Для побудови потрібно не менше 40 спостережень
Формалізована і докладно розроблена методика	Неадаптивність до нових даних
	Побудова задовільної моделі вимагає великих витрат ресурсів і часу
Прогнози на малий термін	Прогнози на середні та великі терміни



МЕТОД ЕКСПОНЕНЦІЙНОГО ЗГЛАДЖУВАННЯ

Переваги	Недоліки
Прогнози на середні терміни	Прогнози на малі та великі терміни
Адаптивність до нових даних	
Простий, ефективний	
Видно проміжні результати	



НЕЙРОМЕРЕЖІ

Переваги	Недоліки
Прогнози на будь-який термін	Важко знайти проблему при допусканні помилок
Адаптивність до нових даних	
Можливість передбачення динамічних процесів	Принцип роботи «чорний ящик»
Можливо доповнити даними з інших статистичних методів та індексів, подаючи на вхід – беручи найкраще	Доволі складний процес емпіричного «навчання» та перевірки



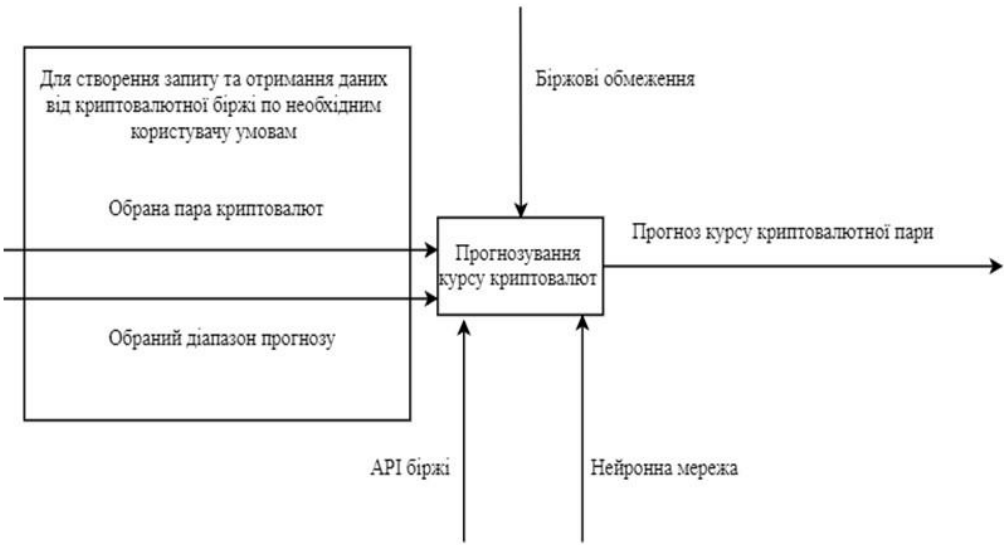
МЕТОД ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

13/29

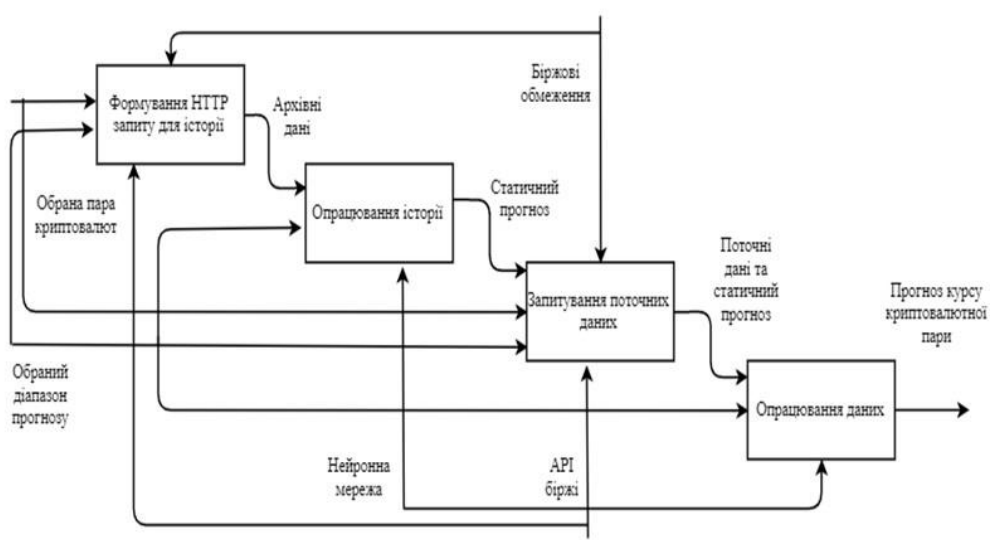
- 1) Збір даних про криптовалюту, які включають в себе:
 - ☐ найменування та ідентифікаційний номер криптовалюти;
 - ☐ поточний курс, та часовий ряд курсу у минулому;
 - ☐ часовий ряд кількості та типу (продаж / купівля) укладених угод;
 - ☐ часовий ряд загальної суми угод у одиницях криптовалюти;
 - ☐ часовий ряд максимумів та мінімумів об'єму угод.
- 2) Подача набору даних паралельно до двох методів для обробки:
 - ☐ методом випадкових процесів та ланцюгів Маркова;
 - ☐ методом обробки ШНМ на архітектурі багат шарового перцептрону навченої по алгоритму зворотнього поширення помилки.
- 3) Передача отриманих часових рядів прогнозу на обробку ШНМ на архітектурі багат шарового перцептрону навченої по алгоритму зворотнього поширення помилки на вибірках прогнозів.
- 4) Видача фінального часового ряда прогнозу.



СТРУКТУРНА МОДЕЛЬ ПРОГРАМНОГО ЗАСОБУ



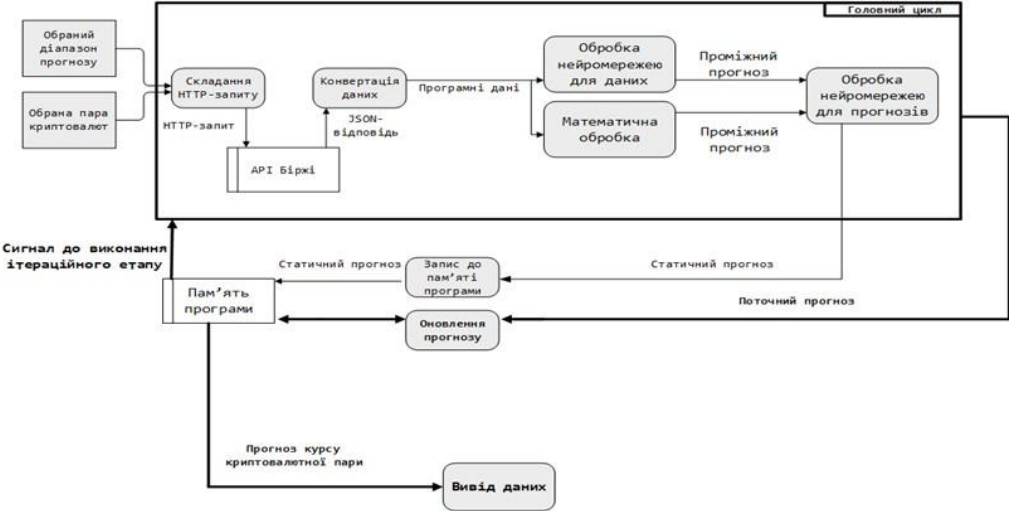
ДЕКОМПОЗИЦІЯ СТРУКТУРНОЇ МОДЕЛІ
ПРОГРАМНОГО ЗАСОБУ В НОТАЦІЇ IDEF0



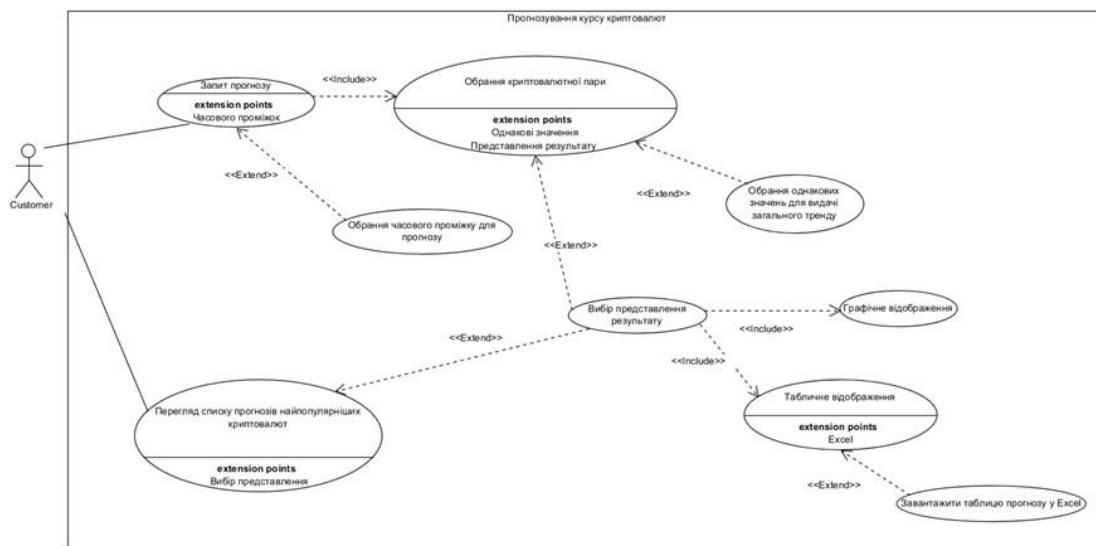
ДЕКОМПОЗИЦІЯ СТРУКТУРНОЇ МОДЕЛІ
ПРОГРАМНОГО ЗАСОБУ В НОТАЦІЇ IDEF3



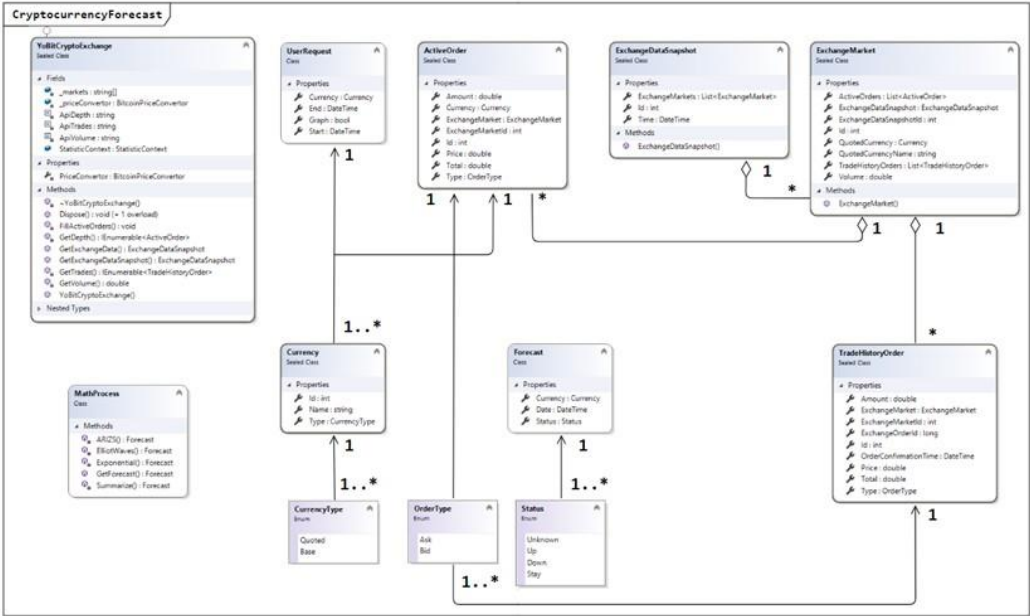
ДЕКОМПОЗИЦІЯ КОНЦЕПТУАЛЬНОЇ МОДЕЛІ
ПРОГРАМНОГО ЗАСОБУ В НОТАЦІЇ DFD



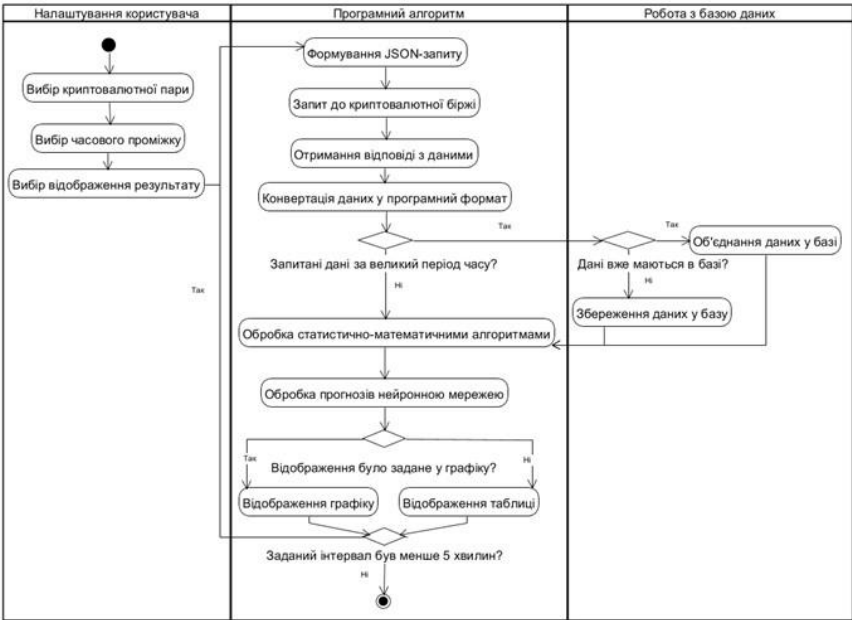
СТРУКТУРНА СХЕМА ВАРІАНТІВ ВИКОРИСТАННЯ



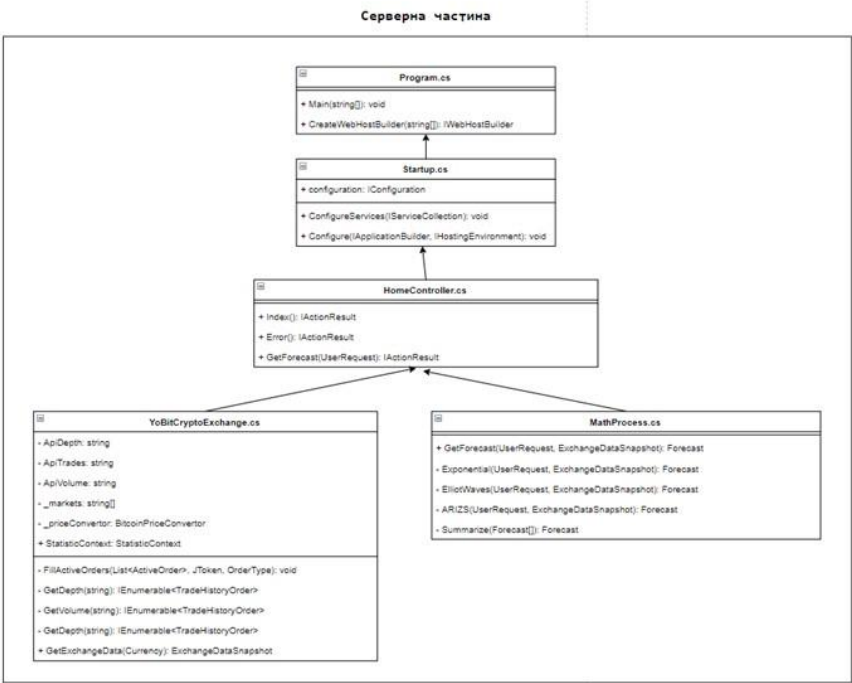
ДІАГРАМА КЛАСІВ



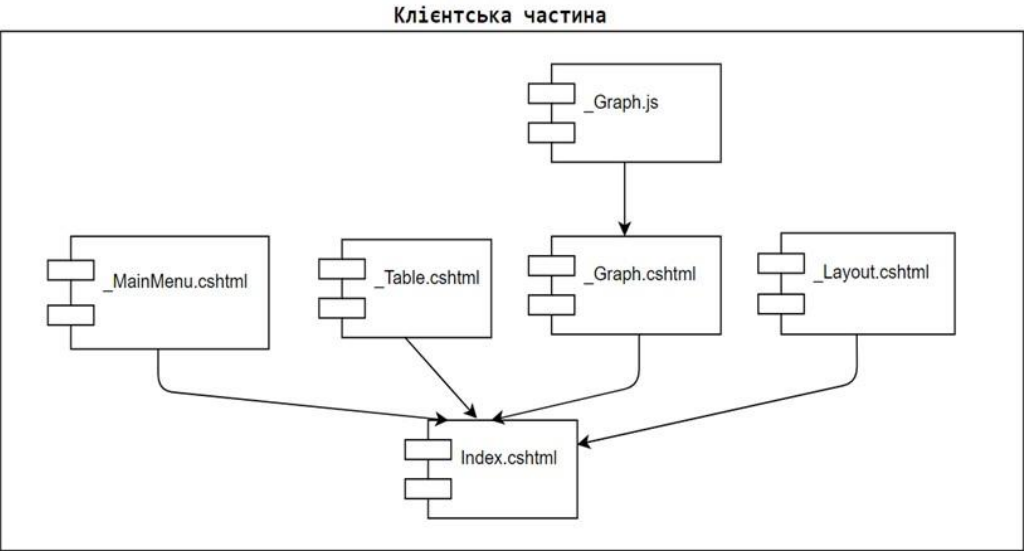
Діаграма діяльності



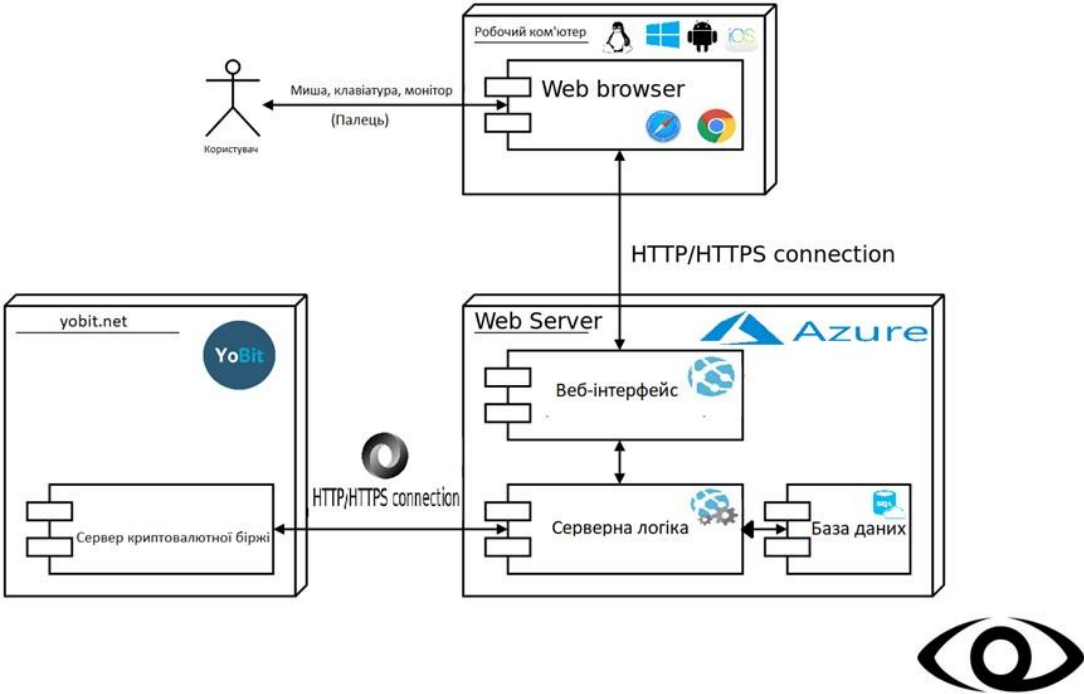
Діаграма компонентів серверної частини



Діаграма компонентів клієнтської частини



Діаграма компонентів клієнтської частини



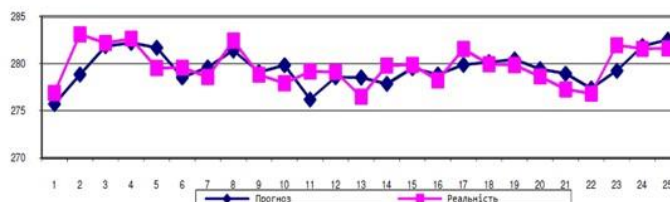
РЕЗУЛЬТАТИ

24/29

ШНМ №1 навчена за алгоритмом зворотнього поширення помилки для обробки часового ряду з даними криптовалютної біржі

❑ Середня помилка прогнозування $E = 0.4487\%$

❑ Точність прогнозу напрямку тренду $T = 52\%$.



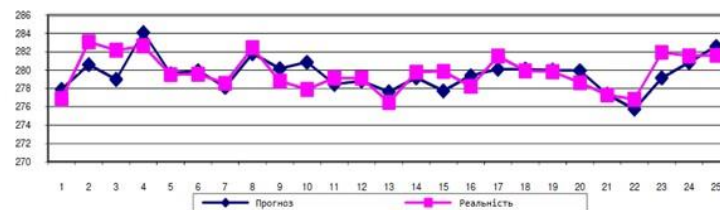
РЕЗУЛЬТАТИ

25/29

ШНМ №2 навчена для комбінування прогнозів на основі вхідних даних з інших прогнозів.

❑ Середня помилка прогнозування $E = 0.4106\%$

❑ Точність прогнозу напрямку тренду $T = 76\%$.



ВЕБ-ДОДАТОК ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

26/29



ВИСНОВКИ

Проаналізовано методи для прогнозування курсу валют, з яких найефективнішими виявилися – методика із застосуванням принципів Ланцюгів Маркова та нейромережі на основі перцептрона.

Було вирішено використати окремий випадок перцептрона – перцептрон Румельхарта для його поєднання з іншими методами у якості навчальних шарів. Щоб у результаті мати аналог фундаменальному аналізу, та наявним методам технічного аналізу.

Як результат було створено метод та програмний засіб на його основі за допомогою на платформі .NET, який збільшив середню точність прогнозу на 4-24% в залежності від типу прогнозу.



СПИСОК ОПУБЛІКОВАНИХ НАУКОВИХ ПРАЦЬ

- [1] Яхін С.Л., Цуркан В.В. Аналіз програмних засобів прогнозування курсу криптовалют. Прикладна математика та комп'ютинг : тези доповідей конференції молодих вчених. (м. Київ, 14 – 16 листопада 2018 р.). Київ, 2018. С. 1-4.
- [2] Яхін С.Л., Цуркан В.В. Концептуальна модель програмного засобу прогнозування курсу криптовалюти. Моделювання та інформаційні технології : збірник наукових праць. 2019. Вип. 86. С. 1-6. – (підготовлено до друку)
- [3] Яхін С.Л., Цуркан В.В. Програмний засіб прогнозування курсу криптовалют. XXXVII науково-технічна конференція молодих вчених та спеціалістів інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України. (м. Київ, 15 травня 2019 р.). Київ, 2019. С. 1.
- [4] Яхін С.Л., Цуркан В.В. Об'єктно-орієнтована модель програмного засобу прогнозування курсу криптовалют. Актуальні питання застосування спеціальних інформаційно-телекомунікаційних систем : тези доповідей конференції студентів, аспірантів, докторантів та молодих учених (м. Київ, 22 – 23 травня 2019 р.). Київ, 2019. С. 1.

Дякую за увагу!



Додаток 2
Текст програми

Лістинг 1. Фрагмент тексту класу YoBitCryptoExchange, що реалізує доступ до API біржі.

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Net;
using System.Threading.Tasks;
using CryptoExchangeService.Statistic.Helpers;
using CryptoExchangeService.Statistic.Models;
using CryptoExchangeService.Statistic.Models.Enums;
using Newtonsoft.Json.Linq;
using static System.Console;
using static System.Globalization.DateTimeStyles;
using static
CryptoExchangeService.Statistic.Models.Enums.CurrencyType;
using static CryptoExchangeService.Statistic.Models.Enums.OrderType;

namespace CryptoExchangeService.Statistic
{
    public sealed class YoBitCryptoExchange : IDisposable
    {
        private const string ApiDepth =
"https://yobit.net/api/3/depth/";

        private const string ApiTrades =
"https://yobit.net/api/3/trades/";

        private const string ApiVolume =
"https://yobit.net/api/3/ticker/";

        private readonly string[] _markets;

        public readonly StatisticContext StatisticContext;

        private BitcoinPriceConvertor _priceConvertor;

        public YoBitCryptoExchange()
        {
            StatisticContext = new StatisticContext();
            _markets = new[] { "btc", "eth", "doge",
"waves", "usd", "run" };
        }

        private BitcoinPriceConvertor PriceConvertor =>
            _priceConvertor ?? (_priceConvertor = new
BitcoinPriceConvertor());
    }
}
```



```

        public ExchangeDataSnapshot GetExchangeDataSnapshot(string
currency)
        {
            var exchangeDataSnapshot = new ExchangeDataSnapshot
            {
                Time = DateTime.Now
            };

            Parallel.ForEach(_markets,
                market =>
                {
                    string pair =
                        $"{currency}_{market}";

                    var exchangeMarket = new
ExchangeMarket
                    {
                        QuotedCurrencyName = market,
                        ActiveOrders =
GetDepth(pair).ToList(),
                        TradeHistoryOrders =
GetTrades(pair).ToList(),
                        Volume =
GetVolume(pair)
                    };

                    lock (exchangeDataSnapshot)
                    {
                        if
                        (exchangeMarket.ActiveOrders.Count > 0 ||
exchangeMarket.TradeHistoryOrders.Count > 0)
                        {
                            exchangeDataSnapshot.ExchangeMarkets.Add(exchangeMarket);
                        }
                    }
                });

            foreach (ExchangeMarket exchangeMarket in
exchangeDataSnapshot.ExchangeMarkets)
            {
                exchangeMarket.QuotedCurrency = StatisticContext
.Currencies
                .SingleOrDefault(c
=> c.Name == exchangeMarket.QuotedCurrencyName &&
c.Type == Quoted) ??
                new Currency
                {

```

```

Name =
exchangeMarket.QuotedCurrencyName,
Type = Quoted
});

exchangeMarket.ActiveOrders.ForEach(ao => ao.Price =
PriceConvertor.ConvertToBTC(ao.Price,
exchangeMarket.QuotedCurrencyName));
}

return exchangeDataSnapshot;
}

private static IEnumerable<TradeHistoryOrder>
GetTrades(string pair)
{
    const string timestamp = "timestamp";
    const string price      = "price";
    const string amount     = "amount";
    const string type       = "type";
    const string ask        = "ask";

    using (var webClient = new WebClient())
    {
        string data = webClient.DownloadString(ApiTrades +
pair);

        try
        {
            JObject jData = JObject.Parse(data);

            return jData[pair]
                .Select(part => new TradeHistoryOrder
                {
                    Price =
part[price].Value<double>(),
                    Amount =
part[amount].Value<double>(),
                    Type =
part[type].ToString() == ask ? Ask : Bid,
                    OrderConfirmationTime =
part[timestamp].UnixTimeStampToDateTime(),
                    ExchangeOrderId =
part[timestamp].Value<long>()
                });
        }
        catch (Exception ex)
        {
            WriteLine($"{ApiDepth}{pair} {ex.Message}");
        }
    }
}

```

```

        return Enumerable.Empty<TradeHistoryOrder>();
    }
}

private static IEnumerable<ActiveOrder> GetDepth(string
pair)
{
    const string asks = "asks";
    const string bids = "bids";
    const string error = "error";

    using (var webClient = new WebClient())
    {
        string data = webClient.DownloadString(ApiDepth +
pair);

        try
        {
            JObject jobject = JObject.Parse(data);

            var activeOrders = new List<ActiveOrder>();

            if (jobject.TryGetValue(error,
StringComparison.InvariantCultureIgnoreCase, out JToken jToken))
            {
                WriteLine($"{ApiDepth}{pair} {jToken}");

                return Enumerable.Empty<ActiveOrder>();
            }

            if (!jobject.TryGetValue(pair,
StringComparison.InvariantCultureIgnoreCase, out jToken))
            {
                WriteLine($"{ApiDepth}{pair} doesn't contain
correct answer");

                return Enumerable.Empty<ActiveOrder>();
            }

            FillActiveOrders(activeOrders,
jToken.Value<JToken>(asks), Ask);
            FillActiveOrders(activeOrders,
jToken.Value<JToken>(bids), Bid);

            return activeOrders;
        }
        catch (Exception ex)
        {
            WriteLine($"{ApiDepth}{pair} {ex.Message}");

```

```

        return Enumerable.Empty<ActiveOrder>();
    }
}

private static double GetVolume(string pair)
{
    const string volume = "vol";

    try
    {
        using (var wc = new WebClient())
        {
            JObject jobject =
JObject.Parse(wc.DownloadString(ApiVolume + pair));

            return
jObject.Value<JToken>(pair).Value<double>(volume);
        }
    }
    catch (Exception ex)
    {
        WriteLine($"{DateTime.Now} {ex.Message} Thrown from
GetVolume method");

        return default(double);
    }
}

private static void FillActiveOrders(List<ActiveOrder>
activeOrders, JToken orderTokens, OrderType orderType)
{
    if (orderTokens != null)
    {
        activeOrders.AddRange(
orderTokens.Select(part => new
ActiveOrder
{
    Price =
part[0].Value<double>(),
    Amount =
part[1].Value<double>(),
    Type = orderType
}));
    }
}

~YoBitCryptoExchange()
{
    Dispose(false);
}

```

```

    }

    #region IDisposable

    private void Dispose(bool disposing)
    {
        if (disposing)
        {
            StatisticContext?.Dispose();
        }
    }

    public void Dispose()
    {
        Dispose(true);
        GC.SuppressFinalize(this);
    }

    #endregion

    private sealed class BitcoinPriceConvertor
    {
        private const string ApiPrice =
        "https://yobit.net/api/3/ticker/eth_btc-doge_btc-waves_btc-btc_usd-
        btc_rur";

        private readonly Dictionary<string, double>
        _currenciesCoefficients;

        internal BitcoinPriceConvertor()
        {
            _currenciesCoefficients = new Dictionary<string,
double>();
            GetBtcPrice();
        }

        private double getAveragePrice(JToken price)
        {
            const double averageCoef = 2.0;

            return (price["low"].Value<double>() +
price["high"].Value<double>()) / averageCoef;
        }

        private static double GetRevertPrice(double price)
        {
            return 1 / price;
        }

        private void GetBtcPrice()
        {

```

```

        try
        {
            using (var wc = new WebClient())
            {
                string data = wc.DownloadString(ApiPrice);

                JObject jdata = JObject.Parse(data);

                _currenciesCoefficients.Add("eth",
getAveragePrice(jdata["eth_btc"]));
                _currenciesCoefficients.Add("doge",
getAveragePrice(jdata["doge_btc"]));
                _currenciesCoefficients.Add("waves",
getAveragePrice(jdata["waves_btc"]));

                _currenciesCoefficients.Add("run",
GetRevertPrice(getAveragePrice(jdata["btc_run"])));
                _currenciesCoefficients.Add("usd",
GetRevertPrice(getAveragePrice(jdata["btc_usd"])));
            }
        }
        catch (Exception ex)
        {
            WriteLine($"{DateTime.Now} {ex.Message} Thrown
from {nameof(PriceConvertor)}");
        }
    }

    public double ConvertToBTC(double price, string
quotedCurrencyName)
    {
        if
(_currenciesCoefficients.TryGetValue(quotedCurrencyName, out double
coef))
        {
            return price * coef;
        }

        return price;
    }
}
}
}
}

```

Лістинг 2. Фрагмент тексту класу Binance, що реалізує доступ до API
запасної біржі.

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Net;
using System.Threading;
using Newtonsoft.Json.Linq;

namespace CryptoExchangeService.Statistic
{
    public static class Binance
    {
        private const string ApiString = "https://api.binance.com/";

        private static readonly Dictionary<string, Method> Methods =
            new Dictionary<string, Method>
            {
                { "ping", new Method { URL = "api/v1/ping"
, Type = "GET", IsPrivate = false } },
                { "time", new Method { URL = "api/v1/time"
, Type = "GET", IsPrivate = false } },
                { "exchangeInfo", new Method { URL =
"api/v1/exchangeInfo", Type = "GET", IsPrivate = false
} },
                { "depth", new Method { URL = "api/v1/depth"
, Type = "GET", IsPrivate = false } },
                { "trades", new Method { URL = "api/v1/trades"
, Type = "GET", IsPrivate = false } },
                { "historicalTrades", new Method { URL =
"api/v1/historicalTrades", Type = "GET", IsPrivate = false
} },
                { "aggTrades", new Method { URL =
"api/v1/aggTrades", Type = "GET", IsPrivate = false
} },
                { "klines", new Method { URL = "api/v1/klines"
, Type = "GET", IsPrivate = false } },
                { "ticker24hr", new Method { URL =
"api/v1/ticker/24hr", Type = "GET", IsPrivate = false
} },
                { "tickerPrice", new Method { URL =
"api/v3/ticker/price", Type = "GET", IsPrivate = false
} },
                { "tickerBookTicker", new Method { URL =
"api/v3/ticker/bookTicker", Type = "GET", IsPrivate = false
} },
                { "createOrder", new Method { URL = "api/v3/order"
, Type = "POST", IsPrivate = true } },
            }
    }
}
```

```

        { "testOrder"          , new Method { URL =
"api/v3/order/test"          , Type = "POST"   , IsPrivate = true
} } ,
        { "orderInfo"         , new Method { URL = "api/v3/order"
, Type = "GET"       , IsPrivate = true } } ,
        { "cancelOrder"       , new Method { URL = "api/v3/order"
, Type = "DELETE", IsPrivate = true } } ,
        { "openOrders"        , new Method { URL =
"api/v3/openOrders"         , Type = "GET"    , IsPrivate = true
} } ,
        { "allOrders"         , new Method { URL =
"api/v3/allOrders"          , Type = "GET"    , IsPrivate = true
} } ,
        { "account"           , new Method { URL =
"api/v3/account"            , Type = "GET"    , IsPrivate = true
} } ,
        { "myTrades"          , new Method { URL =
"api/v3/myTrades"           , Type = "GET"    , IsPrivate = true
} } ,
        { "depositAddress"    , new Method { URL =
"/wapi/v3/depositAddress.html" , Type = "GET"    , IsPrivate = true
} } ,
        { "withdraw"          , new Method { URL =
"/wapi/v3/withdraw.html"     , Type = "POST"   , IsPrivate = true
} } ,
        { "depositHistory"    , new Method { URL =
"/wapi/v3/depositHistory.html" , Type = "GET"    , IsPrivate = true
} } ,
        { "withdrawHistory"   , new Method { URL =
"/wapi/v3/withdrawHistory.html" , Type = "GET"    , IsPrivate = true
} } ,
        { "withdrawFee"       , new Method { URL =
"/wapi/v3/withdrawFee.html"   , Type = "GET"    , IsPrivate = true
} } ,
        { "accountStatus"     , new Method { URL =
"/wapi/v3/accountStatus.html" , Type = "GET"    , IsPrivate = true
} } ,
        { "systemStatus"      , new Method { URL =
"/wapi/v3/systemStatus.html"   , Type = "GET"    , IsPrivate = true
} }
    };

```

```

    /// <summary>
    /// </summary>
    /// <param name="interval">Interval of getting new data for
comparsion in minutes</param>
    /// <param name="period">Comaprson period in minutes,
default it's 24 hours</param>
    /// <param name="pair"></param>
    public static string Klines(int interval = 1, int period =
5, string pair = "BNBBTC")

```



```

{
    const string command          = "klines";
    string          intervalString = $"{interval}m";

    //https://api.binance.com/api/v1/klines?symbol=LTCBTC&interval=5m
    string apiUrl =
    $"{ApiString}{Methods[command].URL}?symbol={pair}&interval={intervalString}";

    string resultDir =
    $"{DateTime.Now.ToShortTimeString().Replace(":", "_")}_{pair}";

    Directory.CreateDirectory(resultDir);

    string path = $"{resultDir}/Oscillation.json";

    using (var wc = new WebClient())
    {
        (long startTimeStamp, decimal startPrice) =
        GetCurrentPrice(apiUrl, wc);

        using (StreamWriter sw = File.CreateText(path))
        {
            sw.WriteLine("{ \"data\":[");

            WriteStamp(sw, startTimeStamp, startPrice);

            sw.WriteLine(",");
        }

        for (var i = 0; i < period; i += interval)
        {
            Thread.Sleep(TimeSpan.FromMinutes(interval));

            (long timeStamp, decimal price) =
            GetCurrentPrice(apiUrl, wc);

            decimal diff = price - startPrice;

            if (Math.Abs(diff / startPrice) <= 0.009m)
            {
                continue;
            }

            using (StreamWriter sw = File.AppendText(path))
            {
                WriteStamp(sw, timeStamp, price, diff, diff
/ startPrice * 100m);

                if (i + interval < period)

```

```

        {
            sw.WriteLine(",");
        }
    }

    startPrice = price;
}

using (StreamWriter sw = File.AppendText(path))
{
    sw.WriteLine("]}");
}

return path;
}

private static void WriteStamp(
    TextWriter sw,
    long        timeStamp,
    decimal     price,
    decimal     diff    = 0,
    decimal     percent = 0)
{
    sw.WriteLine("{");
    sw.WriteLine($"\"time\": {timeStamp.ToDateTime()}\",");
    sw.WriteLine($"\"price\": {price},");
    sw.WriteLine($"\"diff\": {diff},");
    sw.WriteLine($"\"diffPercent\": \"{percent:P}\"");
    sw.WriteLine("}");
}

private static (long timeStamp, decimal price)
GetCurrentPrice(string apiUrl, WebClient wc)
{
    string data = wc.DownloadString(apiUrl);

    int lastKlineStartIndex = data.LastIndexOf('[') + 1;
    int lastKlineEndIndex   = data.LastIndexOf(']') - 1;

    string lastKlineString =
data.Substring(lastKlineStartIndex, lastKlineEndIndex -
lastKlineStartIndex);
    string[] lastKlineValues = lastKlineString.Split(',');

    long.TryParse(lastKlineValues[0], out long timeStamp);
    decimal.TryParse(lastKlineValues[1].Substring(1,
lastKlineValues[1].Length - 2), out decimal price);

    return (timeStamp, price);
}

```

```

    }

    private static string ToDateTime(this long unixTimeStamp)
    {
        var dtDateTime = new DateTime(1970, 1, 1, 0, 0, 0, 0,
DateTimeKind.Utc);

        dtDateTime =
dtDateTime.AddMilliseconds(unixTimeStamp).ToLocalTime();

        return
dtDateTime.ToString(CultureInfo.InvariantCulture);
    }

    public static void ResultsOnScreen(string path)
    {
        string data = File.ReadAllText(path);

        JObject jData = JObject.Parse(data);

        var lastPrice =
jData["data"].Last["price"].Value<decimal>();
        var firstPrice =
jData["data"].First["price"].Value<decimal>();

        decimal diff = lastPrice - firstPrice;
        decimal diffInPercent = diff / firstPrice * 100m;

        using (StreamWriter sw = File.AppendText(path))
        {
            sw.WriteLine($"Oscillation count =
{jData["data"].Count() - 1}");
            sw.WriteLine($"Diff between first and last = {diff}
({diffInPercent:P})");
        }

        Console.WriteLine($"Oscillation count =
{jData["data"].Count() - 1}");
        Console.WriteLine($"Diff between first and last = {diff}
({diffInPercent:P})");
    }

    private sealed class Method
    {
        public bool IsPrivate;
        private string type;
        public string URL;

        public string Type { get => type; set => type = value; }
    }

```

```

    }
}

//https://api.binance.com/api/v1/ticker/24hr?symbol=BNBBTC
/*
* {
    "symbol": "BNBBTC", // пара
    "priceChange": "-94.99999800", // изменение цены за сутки
    "priceChangePercent": "-95.960", // изменение цены за сутки %
    "weightedAvgPrice": "0.29628482", //Средневзвешенная цена
    "prevClosePrice": "0.10002000", // Предыдущая цена закрытия
    "lastPrice": "4.00000200", // Последняя цена
    "lastQty": "200.00000000", // Последний объем
    "bidPrice": "4.00000000", // Цена покупки
    "askPrice": "4.00000200", // Цена продажи
    "openPrice": "99.00000000", // Цена открытия
    "highPrice": "100.00000000", // Самая высокая цена
    "lowPrice": "0.10000000", // Самая низкая цена
    "volume": "8913.30000000", // Объем торгов базовой валюты
    "quoteVolume": "15.30000000", // Объем торгов котируемой
    "openTime": 1499783499040, // Время открытия
    "closeTime": 1499869899040, // Время закрытия
    "fristId": 28385, // Id первой сделки
    "lastId": 28460, // Id последней сделки
    "count": 76 // Кол-во сделок
}
*/

```

Лістинг 3. Фрагмент тексту класу ExchangeMarket, який є основною одиницею сукупності даних отриманих з біржі

```

using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace CryptoExchangeService.Statistic.Models
{
    public sealed class ExchangeMarket
    {
        public ExchangeMarket()
        {
            ActiveOrders = new List<ActiveOrder>();
            TradeHistoryOrders = new List<TradeHistoryOrder>();
        }

        [Key]
        public int Id { get; set; }
    }
}

```

```

        [Display(Name = "Volume")]
        [Required(ErrorMessage = "{0} is required")]
        public double Volume { get; set; }

        [Display(Name = "Quoted Currency")]
        [Required(ErrorMessage = "{0} is required")]
        public Currency QuotedCurrency { get; set; }

        [Display(Name = "Active orders")]
        public List<ActiveOrder> ActiveOrders { get; set; }

        [Display(Name = "Trade history orders")]
        public List<TradeHistoryOrder> TradeHistoryOrders { get;
set; }

        public int ExchangeDataSnapshotId { get; set; }
        public ExchangeDataSnapshot ExchangeDataSnapshot { get; set;
}

        [NotMapped]
        public string QuotedCurrencyName { get; set; }
    }
}

```

Лістинг 4. Фрагмент тексту класу DataController, який є керує обміном інформації між користувачем через представлення, та сервером з обробкою даних

```

using CryptocurrencyForecasting.WebJob.Models;
using CryptocurrencyForecasting.WebUI.Contracts;
using Microsoft.AspNetCore.Mvc;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Auth;
using Microsoft.WindowsAzure.Storage.Table;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace CryptoExchangeService.Statistic.Controllers
{
    [Route("api/[controller]")]
    public class DataController : Controller
    {
        [HttpGet("[action]")]
        public async Task<IEnumerable<FeedContract>> All()
        {

```

```

        var storageAccount = new CloudStorageAccount(new
StorageCredentials("cryptforecast",
"c5CD/ep7+riweoGQit7DKucCJw2Gy42HxcGA++aS00CxMp1GZXxGFpyenORXztC6LpU
NZgxc/ltcgdNGRvrQEA=="), true);

        Uri uri = storageAccount.TableStorageUri.PrimaryUri;
        StorageCredentials cred = storageAccount.Credentials;
        // Create the table client.

        CloudTableClient tableClientWithSAS =
storageAccount.CreateCloudTableClient();
        var query = new TableQuery<Feed>();

        CloudTable table =
tableClientWithSAS.GetTableReference("feedTest");

        TableQuerySegment<Feed> items = await
table.ExecuteQuerySegmentedAsync(query, null);
        IEnumerable<IGrouping<DateTime, Feed>> group =
items.OrderBy(i => i.Date).GroupBy(i => i.Date.Date);

        var list = new List<Feed>();

        foreach (IGrouping<DateTime, Feed> item in group)
        {
            list.Add(item.First());
        }

        var json = JsonConvert.SerializeObject(list);

        return
JsonConvert.DeserializeObject<List<FeedContract>>(json);
    }

    [HttpGet("[action]")]
    public async Task<IEnumerable<CalcResult>> GetCalc()
    {
        var storageAccount = new CloudStorageAccount(new
StorageCredentials("cryptforecast",
"c5CD/ep7+riweoGQit7DKucCJw2Gy42HxcGA++aS00CxMp1GZXxGFpyenORXztC6LpU
NZgxc/ltcgdNGRvrQEA=="), true);

        var uri = storageAccount.TableStorageUri.PrimaryUri;
        var cred = storageAccount.Credentials;
        // Create the table client.

        CloudTableClient tableClientWithSAS =
storageAccount.CreateCloudTableClient();
        var query = new TableQuery<CalcResult>();

```

```

        CloudTable table =
tableClientWithSAS.GetTableReference("calcFeed");

        TableQuerySegment<CalcResult> items = await
table.ExecuteQuerySegmentedAsync(query, null);

        return items.OrderBy(i=>i.Date);
    }

    public class WeatherForecast
    {
        public string DateFormatted { get; set; }

        public int TemperatureC { get; set; }

        public string Summary { get; set; }

        public int TemperatureF
        {
            get
            {
                return 32 + (int)(TemperatureC / 0.5556);
            }
        }
    }
}

```